NATIONAL BUREAU OF STANDARDS REPORT

10 435

A SEARCH AND RESCUE SIMULATION MODEL FOR THE UNITED STATES COAST GUARD

Appendix A

FLOW CHARTS FOR PROGRAMMER LEVEL DOCUMENTATION

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards [1] was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation's central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in four broad program areas. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology.

The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Center for Radiation Research, the Center for Computer Sciences and Technology, and the Office for Information Programs.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of an Office of Measurement Services and the following technical divisions:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic and Molecular Physics—Radio Physics [2]—Radio Engineering [2]—Time and Frequency [2]—Astrophysics [2]—Cryogenics.[2]

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to improved methods of measurement standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; develops, produces, and distributes standard reference materials; relates the physical and chemical properties of materials to their behavior and their interaction with their environments; and provides advisory and research services to other Government agencies. The Institute consists of an Office of Standard Reference Materials and the following divisions:

Analytical Chemistry—Polymers—Metallurgy—Inorganic Materials—Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides technical services to promote the use of available technology and to facilitate technological innovation in industry and Government; cooperates with public and private organizations in the development of technological standards, and test methodologies; and provides advisory and research services for Federal, state, and local government agencies. The Institute consists of the following technical divisions and offices:

Engineering Standards—Weights and Measures — Invention and Innovation — Vehicle Systems Research—Product Evaluation—Building Research—Instrument Shops—Measurement Engineering—Electronic Technology—Technical Analysis.

THE CENTER FOR RADIATION RESEARCH engages in research, measurement, and application of radiation to the solution of Bureau mission problems and the problems of other agencies and institutions. The Center consists of the following divisions:

Reactor Radiation—Linac Radiation—Nuclear Radiation—Applied Radiation.

THE CENTER FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides technical services designed to aid Government agencies in the selection, acquisition, and effective use of automatic data processing equipment; and serves as the principal focus for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Center consists of the following offices and divisions:

Information Processing Standards—Computer Information — Computer Services — Systems Development—Information Processing Technology.

THE OFFICE FOR INFORMATION PROGRAMS promotes optimum dissemination and accessibility of scientific information generated within NBS and other agencies of the Federal government; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System, and provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data—Clearinghouse for Federal Scientific and Technical Information [3]—Office of Technical Information and Publications—Library—Office of Public Information—Office of International Relations.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.
[2] Located at Boulder, Colorado 80302.
[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.

NATIONAL BUREAU OF STANDARDS REPORT

10 435

A SEARCH AND RESCUE SIMULATION MODEL FOR THE UNITED STATES COAST GUARD

Appendix A

FLOW CHARTS FOR PROGRAMMER LEVEL DOCUMENTATION

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

PREFACE

This volume is one of a series which documents a Search and Rescue Simulation Model for the United States Coast Guard. The material reported in this documentation was developed by an interdisciplinary team at the National Bureau of Standards with representation from the U.S. Coast Guard under MIPR Z-70099-0-01935.

The complete documentation is comprised of the following:

Volume I Executive Level Documentation

Volume II Analyst Level Documentation

Volume III Programmer Level Documentation for "PREPROCESSOR"

Volume IV Programmer Level Documentation for "OPSIM"

Volume V Programmer Level Documentation for "POSTPROCESSOR"

Appendix A Flow Charts for Programmer Level Documentation

Appendix B Program Listings for Programmer Level Documentation

The study was initially conducted under the supervision of Martin J. Aronoff; subsequently efforts were supervised by Richard T. Penn, Jr. Technical Project Leadership was supplied throughout the project by Stephen S. Karp. Other participants from the National Bureau of Standards Technical Analysis Division included the following:

Susan S. Chamberlin
Linda K. Cummings
Mary Jane Duberg
William Elliott, III
Walter G. Leight
Joel Levy
Elizabeth E. Leyendecker
Marcia D. Maltese
Patsy L.B. Saunders*
Wayne A. Steele
Michael R. Vogt
Arnold L. Weber

Valuable advice was received from Alan J. Goldman* and Prof. Gustave J. Rath of Northwestern University.

U.S. Coast Guard participants included:

Paul D'Zmura	Gerald L. Underwood
Thomas T. Matteson	Robert R. Wells

Support services were furnished by the following members of the NBS Technical Analysis Divison:

Mary M. Abbott	Frances E. Jones
Theresa I. Conrad	Lucinda I. Farrell

* Staff members of the NBS Applied Mathematics Divison

TABLE OF CONTENTS

Page

I. PREPROCESSOR

PCP 1

NEEDS 9

NUCASE 11

READ 15

FIELD 17

DEMGEN 19

HIST 24

SELECT 27

II. OPSIM

START 28

NUCRU 2

OPSIM 31

SRAS 33

CRES 34

VEC 37

OSET 39

RESAP 40

ROCA 46

MRAS 47

NOTIF 48

TOW 49

DTD 50

SRCH 51

page

NOTE 53

SASS 54

SSS 56

READY. 60

ARSCH 61

COMPL 62

SSET. 63

FUEL 64

HOMEF. 65

SNDBK. 66

SRISE. 68

XSET 69

SERVE. 70

DELAY. 72

ARVSN. 73

STATS. 76

SRCHF. 77

ONSCN. 78

COVER. 80

CHEKN. 81

RETN 82

TERM 83

HOME 85

SAQ 86

QUEUE. 88

EXQ 90

page

STNBY 94

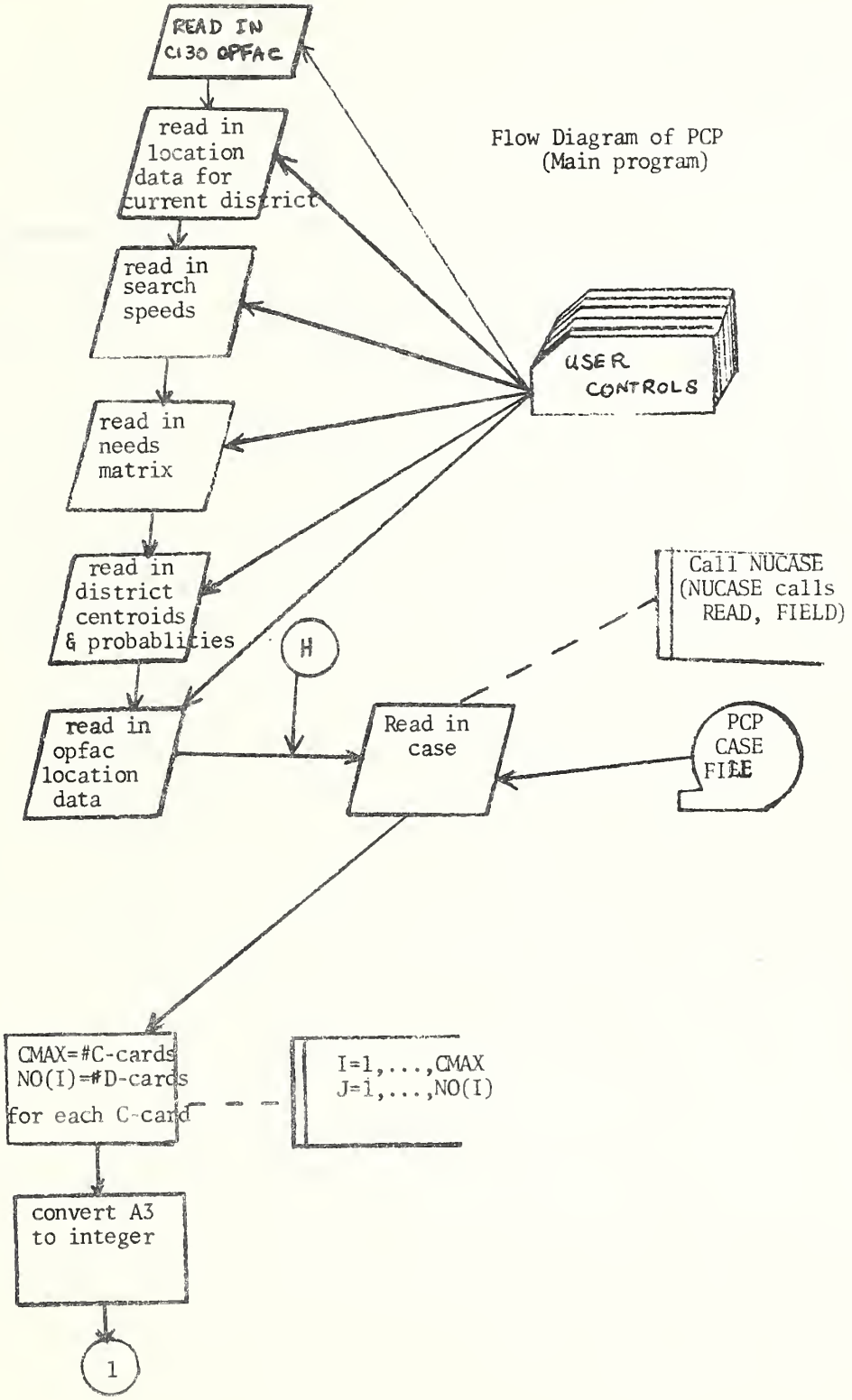
SVQUE 98

WRECK 99

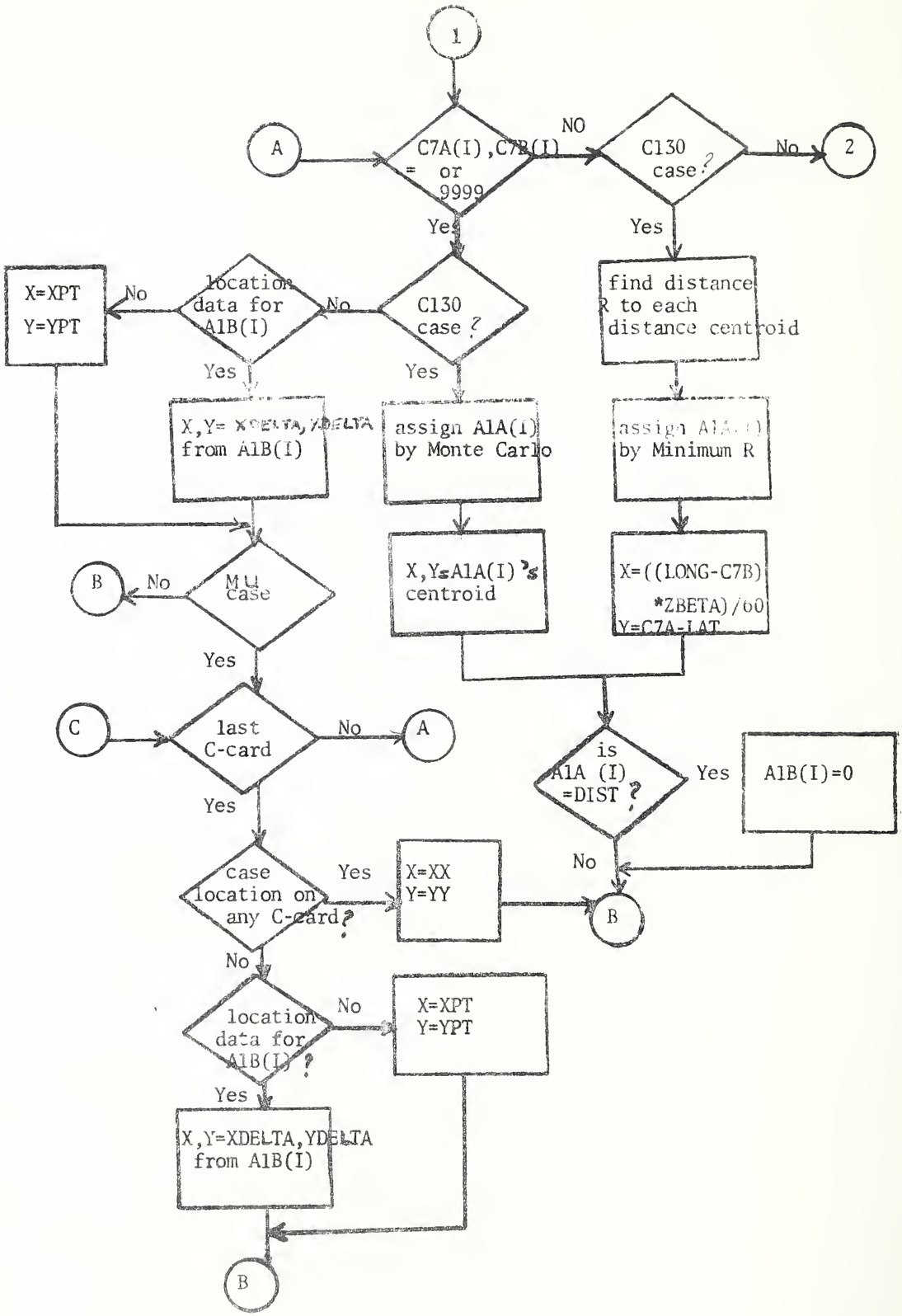
ENDSIM. 100

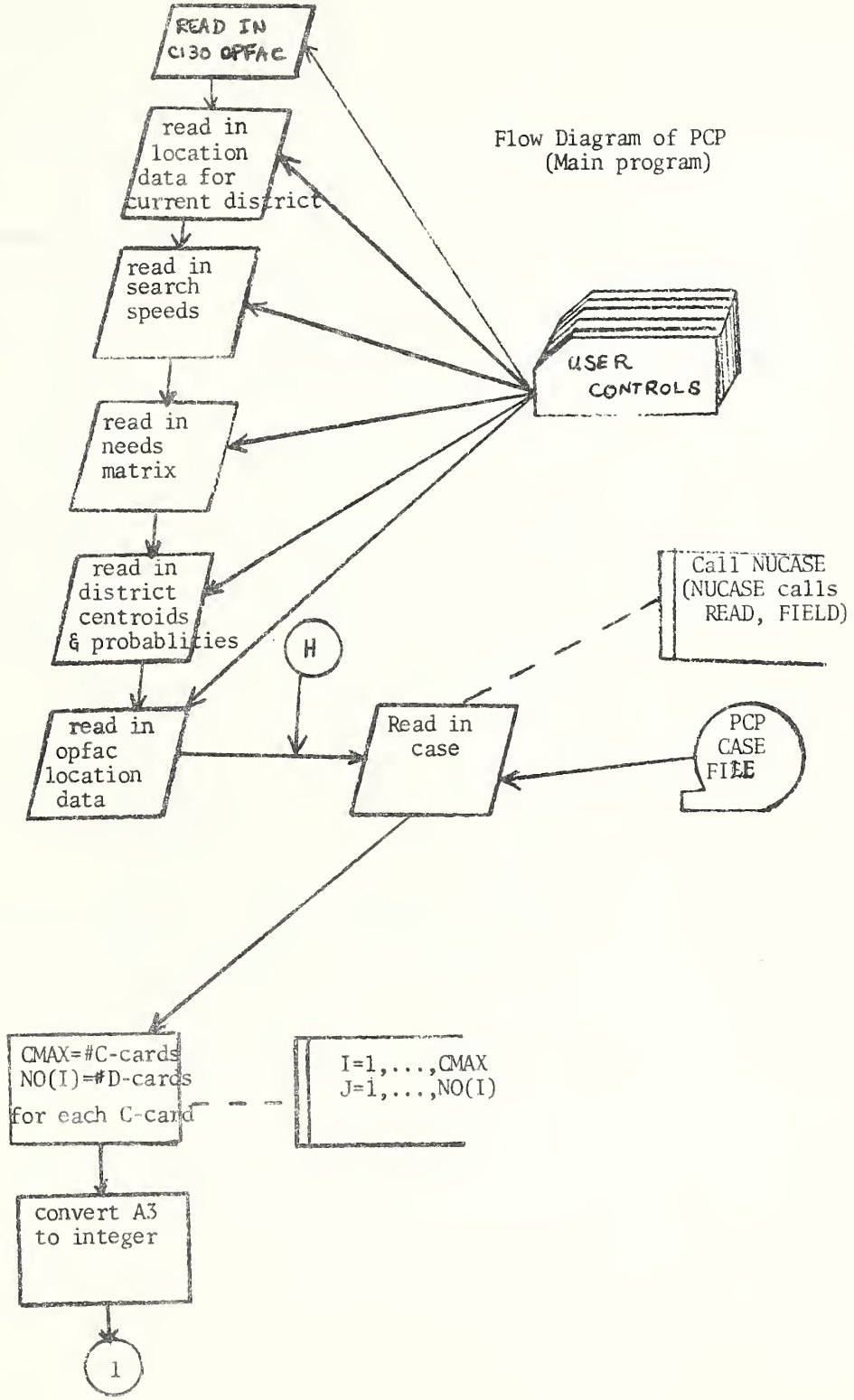
DRIVE 104

JUMPER. 105

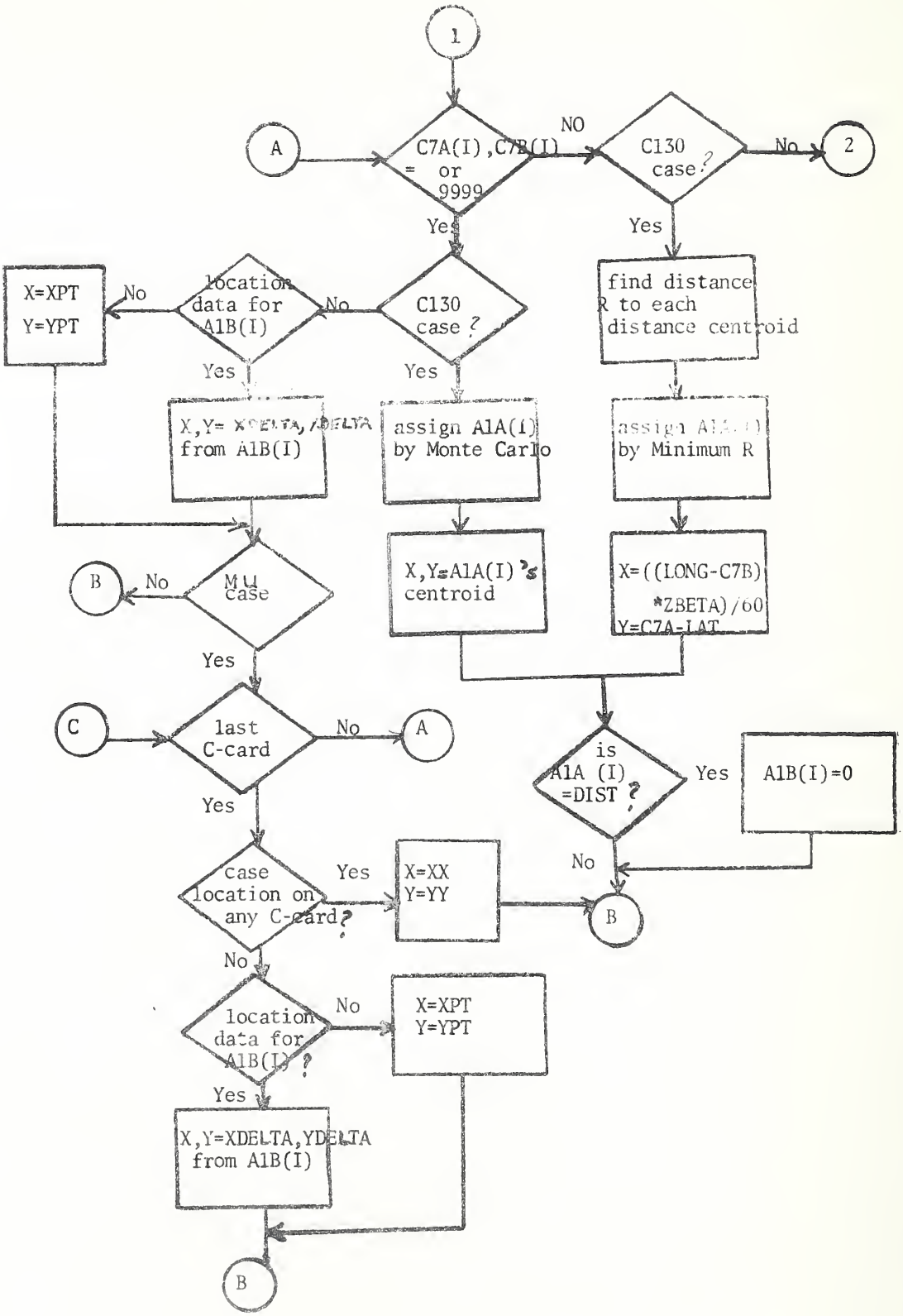


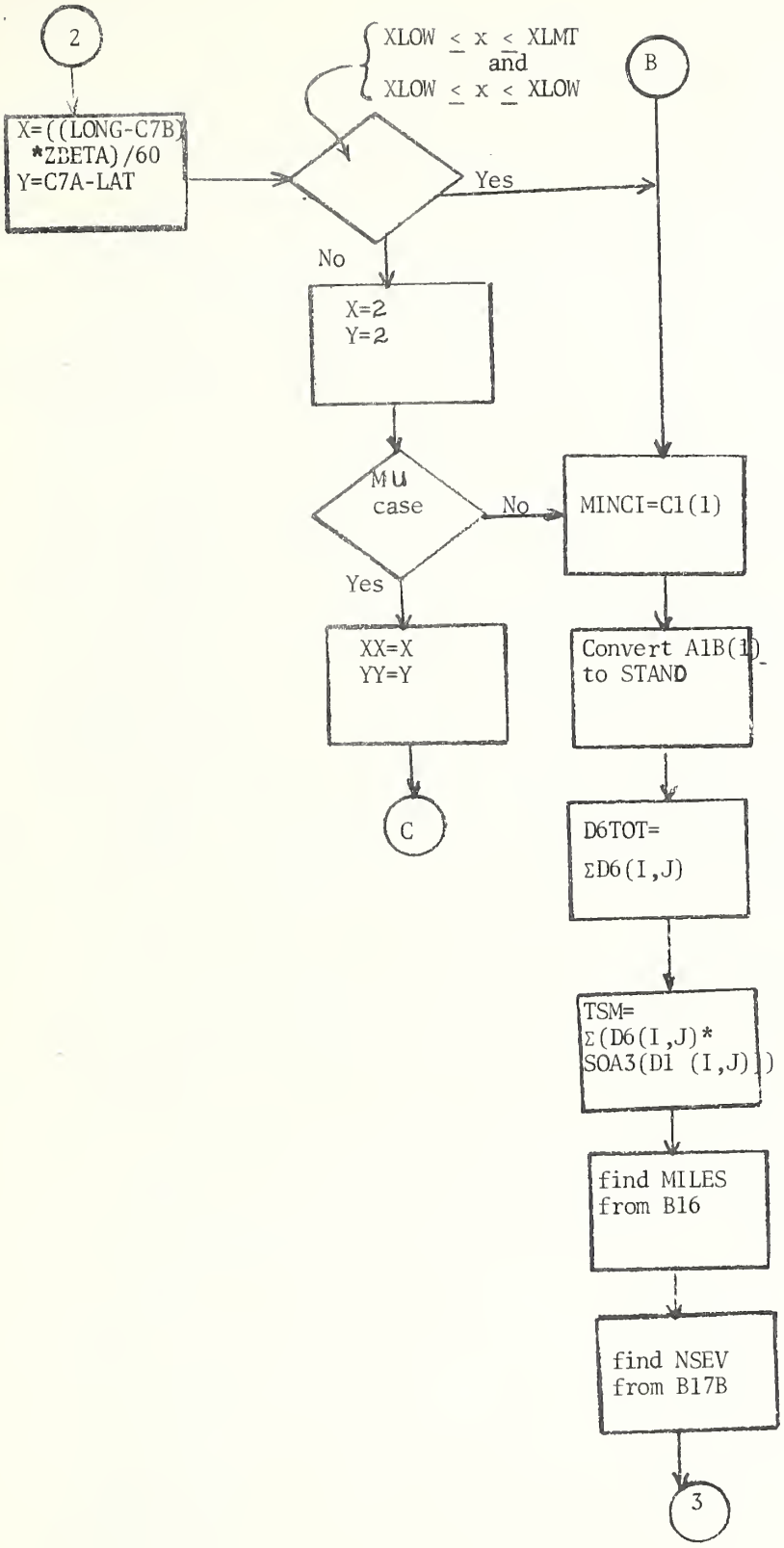
Flow Diagram of PCP
(Main program)

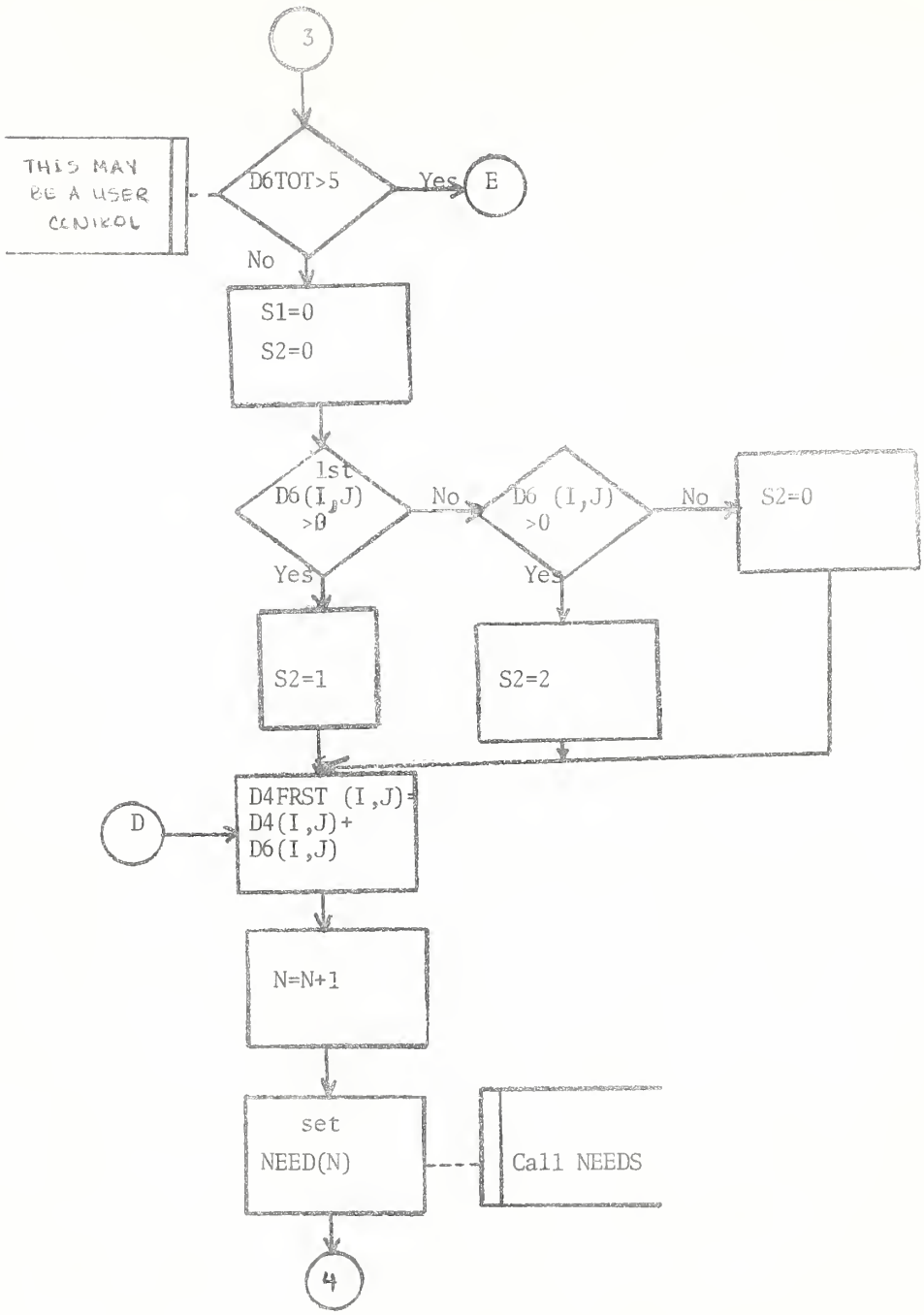

1
A
C7A(I),C7B(I) = or 9999
NO
C130 case?
No
2
Yes
Yes
find distance R to each distance centroid
location data for A1B(I)
No
C130 case?
No
X=XPT
Y=YPT
Yes
Yes
X,Y= XDELTA,YDELTA from A1B(I)
assign A1A(I) by Monte Carlo
assign A1A(I) by Minimum R
X,Y=A1A(I)'s centroid
X=((LONG-C7B) *ZBETA)/60
Y=C7A-LAT
B
No
MU case
Yes
C
last C-card
No
A
Yes
is A1A(I) =DIST?
Yes
A1B(I)=0
No
case location on any C-card?
Yes
X=XX
Y=YY
B
No
location data for A1B(I)?
No
X=XPT
Y=YPT
Yes
X,Y=XDELTA,YDELTA from A1B(I)
B

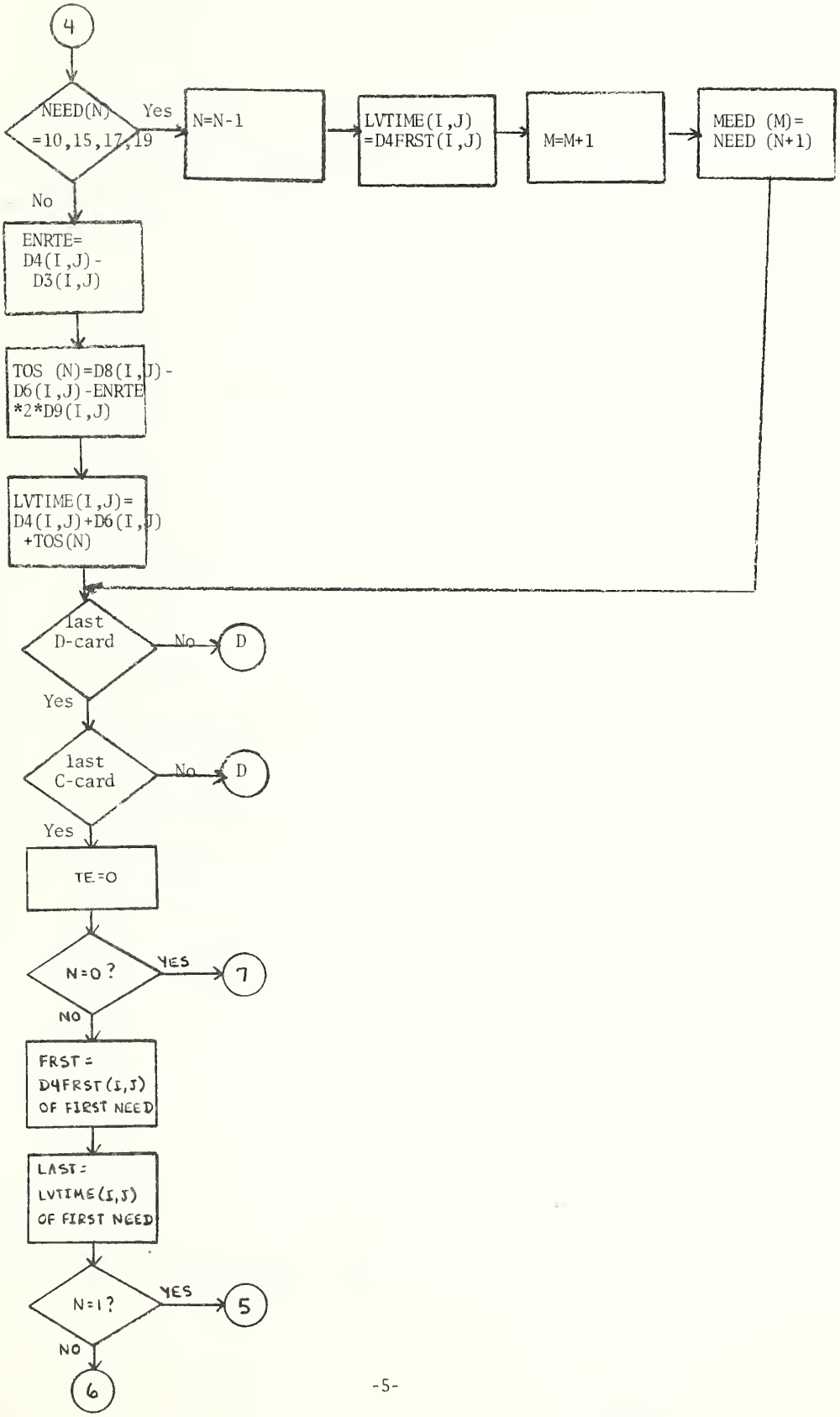


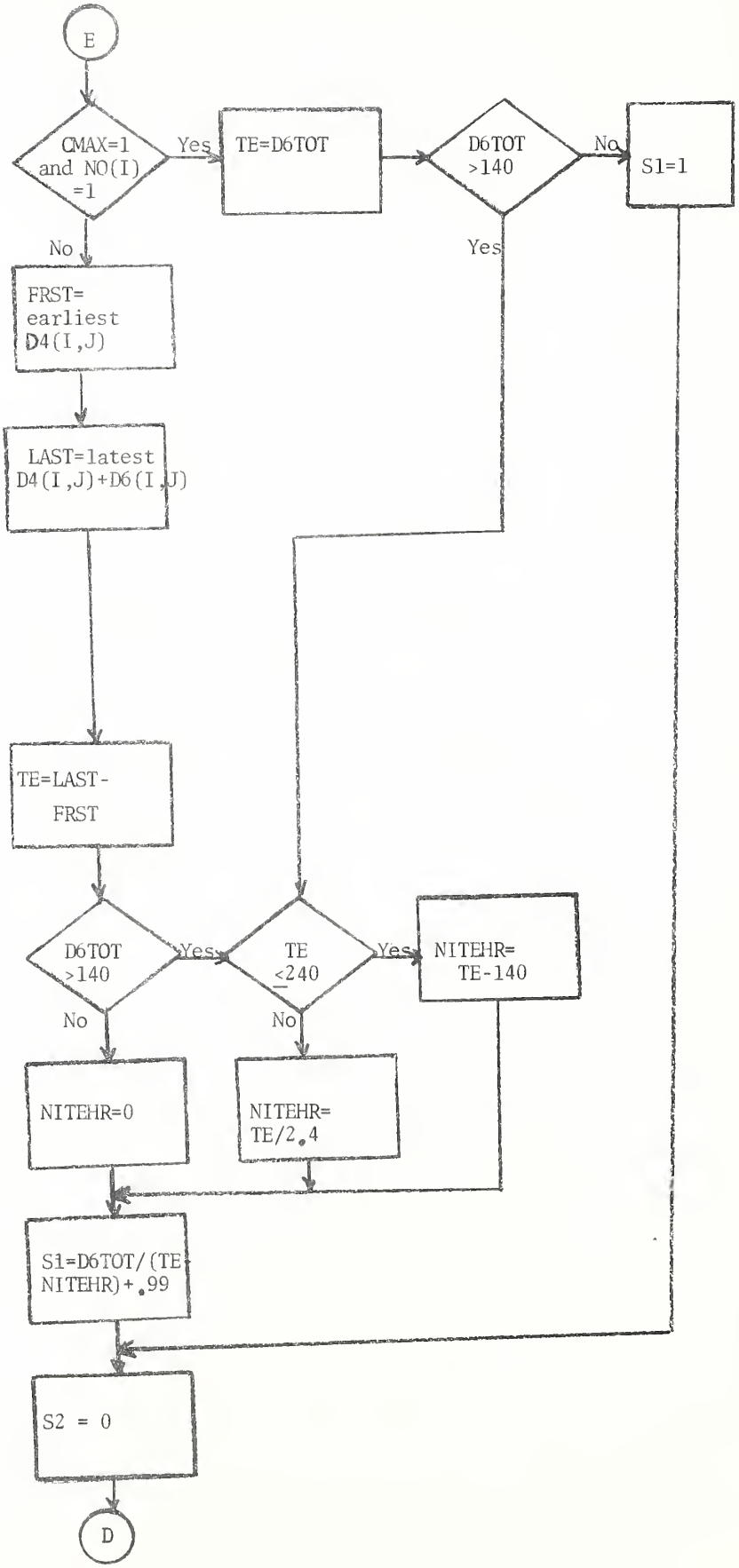
Flow Diagram of PCP
(Main program)

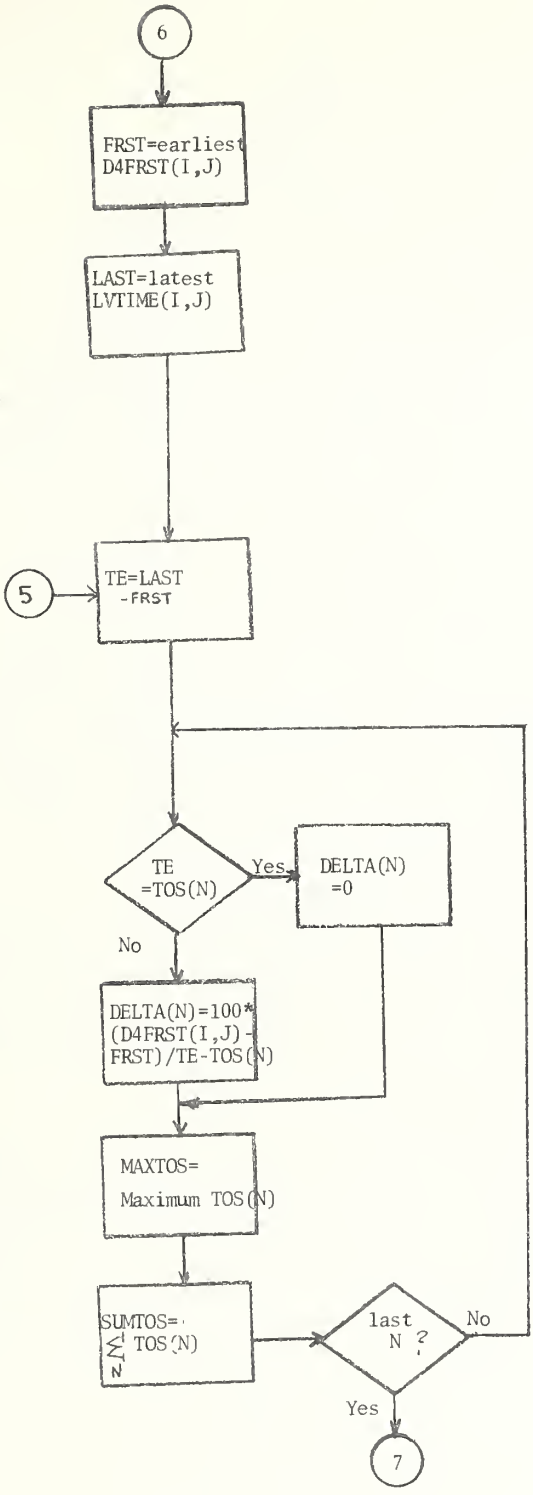

1
A
C7A(I),C7B(I) = or 9999
NO
C130 case?
No
2
Yes
Yes
X=XPT
Y=YPT
No
location data for A1B(I)
No
C130 case?
find distance R to each distance centroid
Yes
Yes
X,Y= XDELTA,YDELTA from A1B(I)
assign A1A(I) by Monte Carlo
assign A1A(I) by Minimum R
B
No
MU case
X,Y=A1A(I)'s centroid
X=((LONG-C7B) *ZBETA)/60
Y=C7A-LAT
Yes
C
last C-card
No
A
is A1A (I) =DIST ?
Yes
A1B(I)=0
Yes
case location on any C-card?
Yes
X=XX
Y=YY
No
B
No
location data for A1B(I) ?
No
X=XPT
Y=YPT
Yes
X,Y=XDELTA,YDELTA from A1B(I)
B

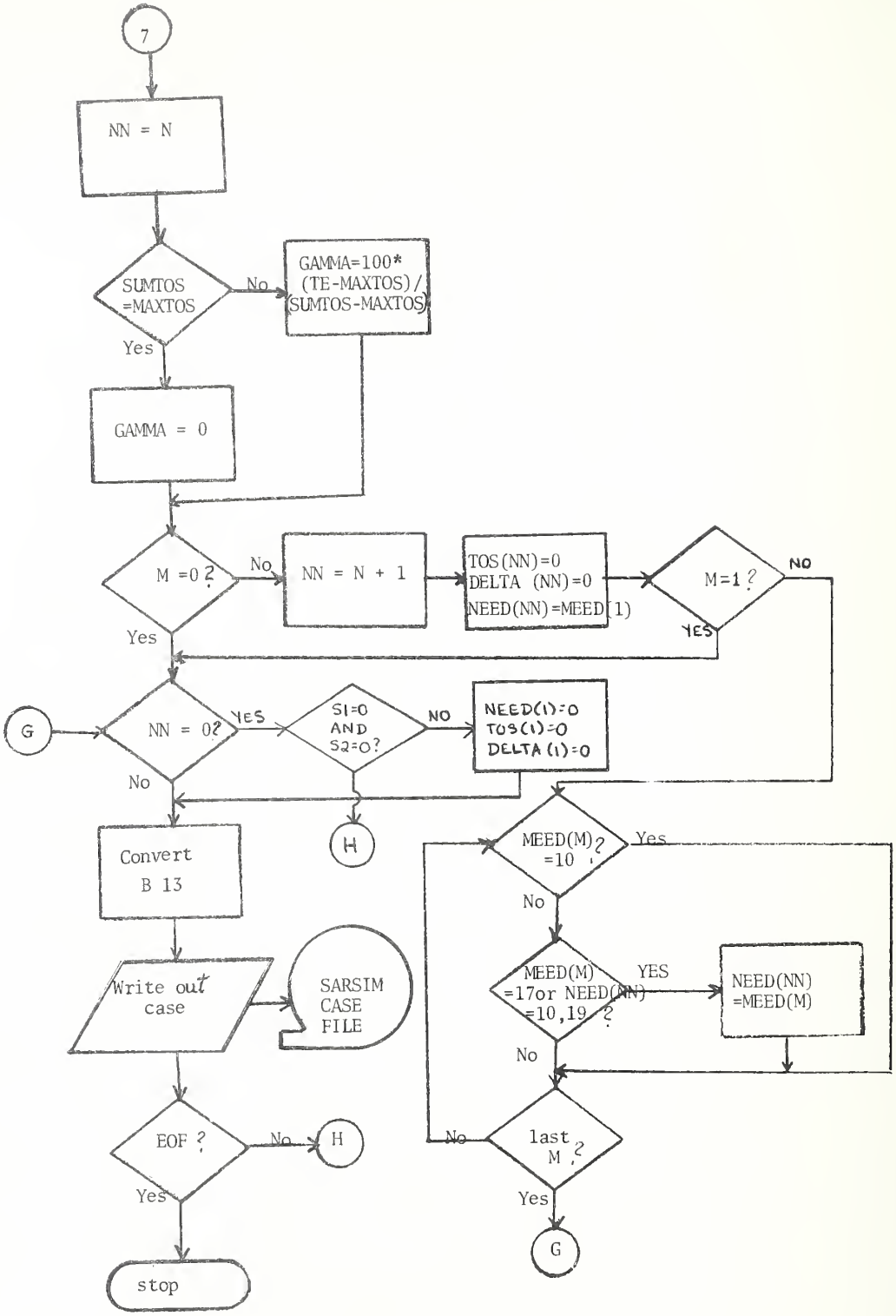

2
X=((LONG-C7B)
*ZBETA)/60
Y=C7A-LAT
XLOW ≤ x ≤ XLMT
and
XLOW ≤ x ≤ XLOW
Yes
No
X=2
Y=2
B
MU
case
No
Yes
MINCI=C1(1)
XX=X
YY=Y
Convert A1B(1)
to STAND
C
D6TOT=
ΣD6(I,J)
TSM=
Σ(D6(I,J)*
SOA3(D1 (I,J))
find MILES
from B16
find NSEV
from B17B
3

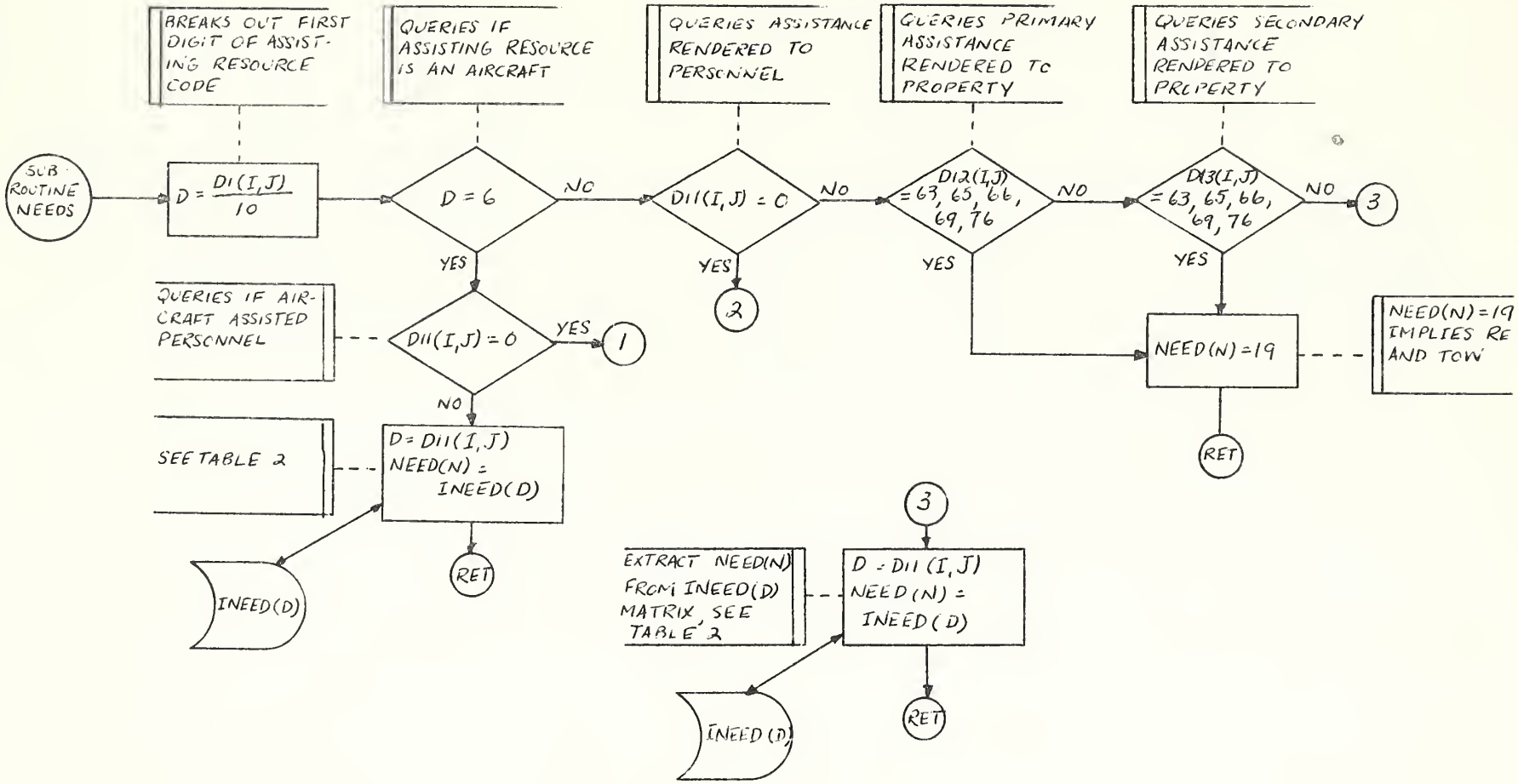

3
THIS MAY BE A USER CONTROL
D6TOT>5
Yes
E
No
S1=0
S2=0
1st D6(I,J) >0
No
D6 (I,J) >0
No
S2=0
Yes
Yes
S2=1
S2=2
D
D4FRST (I,J)= D4(I,J)+ D6(I,J)
N=N+1
set NEED(N)
Call NEEDS
4

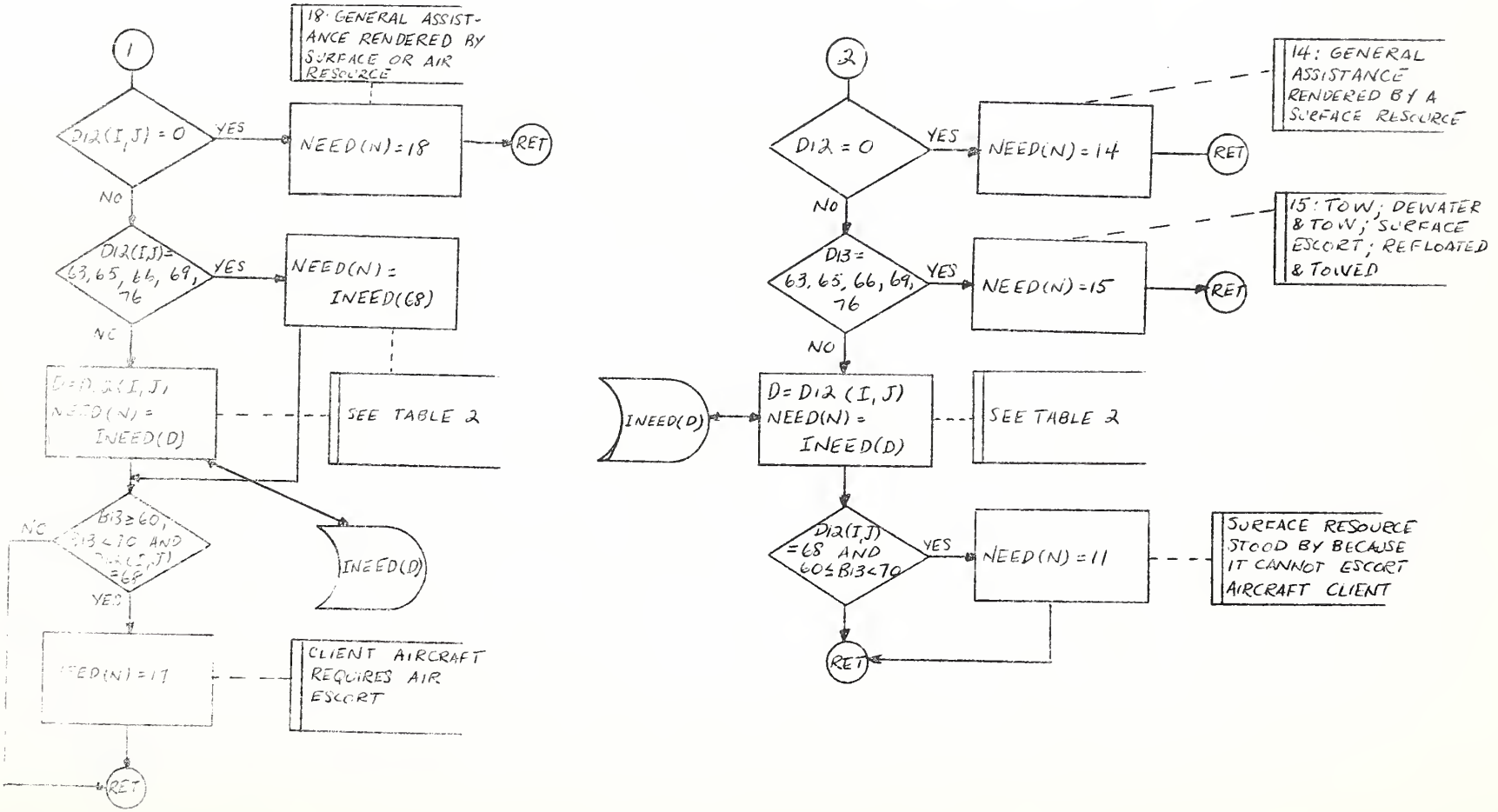

4
NEED(N) =10,15,17,19
Yes
N=N-1
LVTIME(I,J) =D4FRST(I,J)
M=M+1
MEED (M)= NEED (N+1)
No
ENRTE= D4(I,J)- D3(I,J)
TOS (N)=D8(I,J)- D6(I,J)-ENRTE *2*D9(I,J)
LVTIME(I,J)= D4(I,J)+D6(I,J) +TOS(N)
last D-card
No
D
Yes
last C-card
No
D
Yes
TE=0
N=0?
YES
7
NO
FRST = D4FRST(I,J) OF FIRST NEED
LAST = LVTIME(I,J) OF FIRST NEED
N=1?
YES
5
NO
6

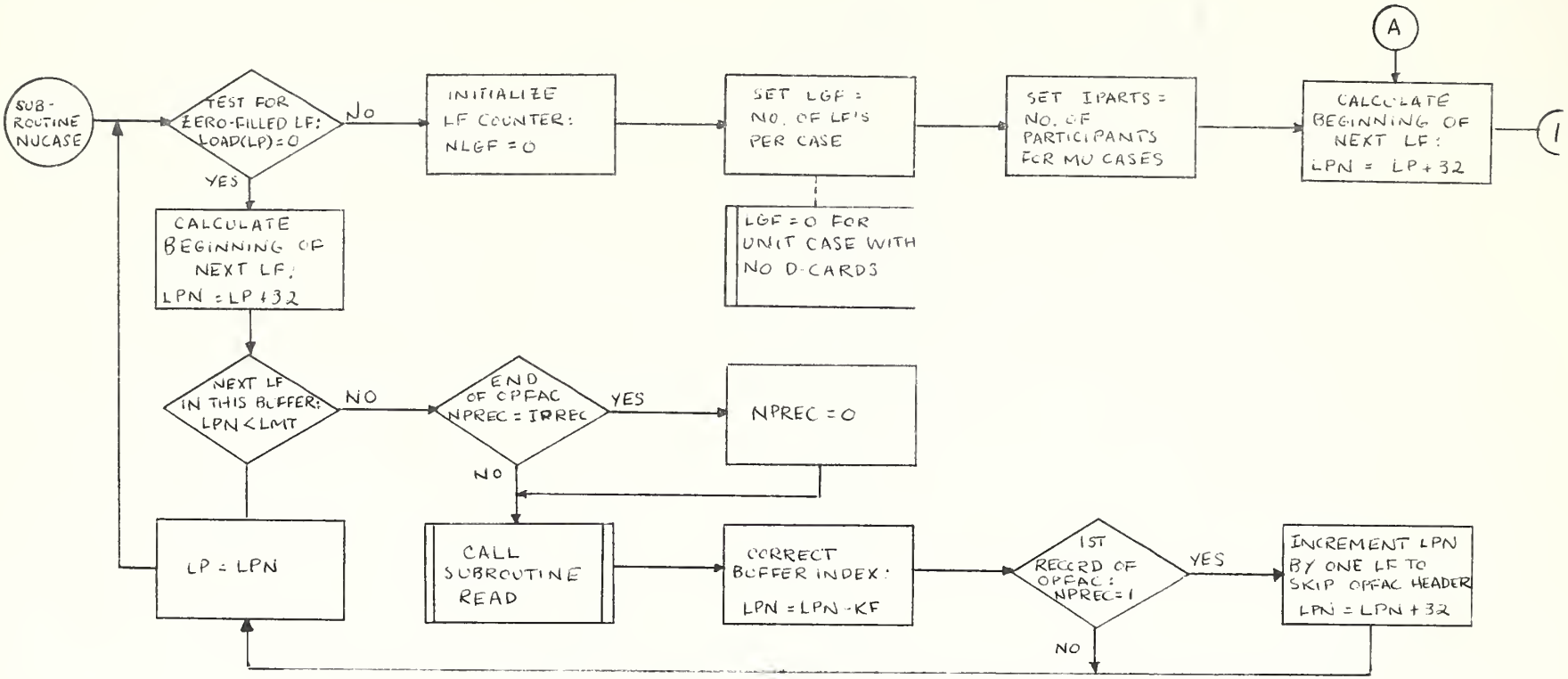

E
CMAX=1 and NO(I) =1
Yes
TE=D6TOT
D6TOT >140
No
S1=1
No
Yes
FRST= earliest D4(I,J)
LAST=latest D4(I,J)+D6(I,J)
TE=LAST- FRST
D6TOT >140
Yes
TE <240
Yes
NITEHR= TE-140
No
No
NITEHR=0
NITEHR= TE/2.4
S1=D6TOT/(TE- NITEHR)+.99
S2 = 0
D


6
FRST=earliest D4FRST(I,J)
LAST=latest LVTIME(I,J)
5
TE=LAST -FRST
TE =TOS(N)
Yes
DELTA(N) =0
No
DELTA(N)=100* (D4FRST(I,J)- FRST)/TE-TOS(N)
MAXTOS= Maximum TOS(N)
SUMTOS= TOS(N)
last N ?
No
Yes
7

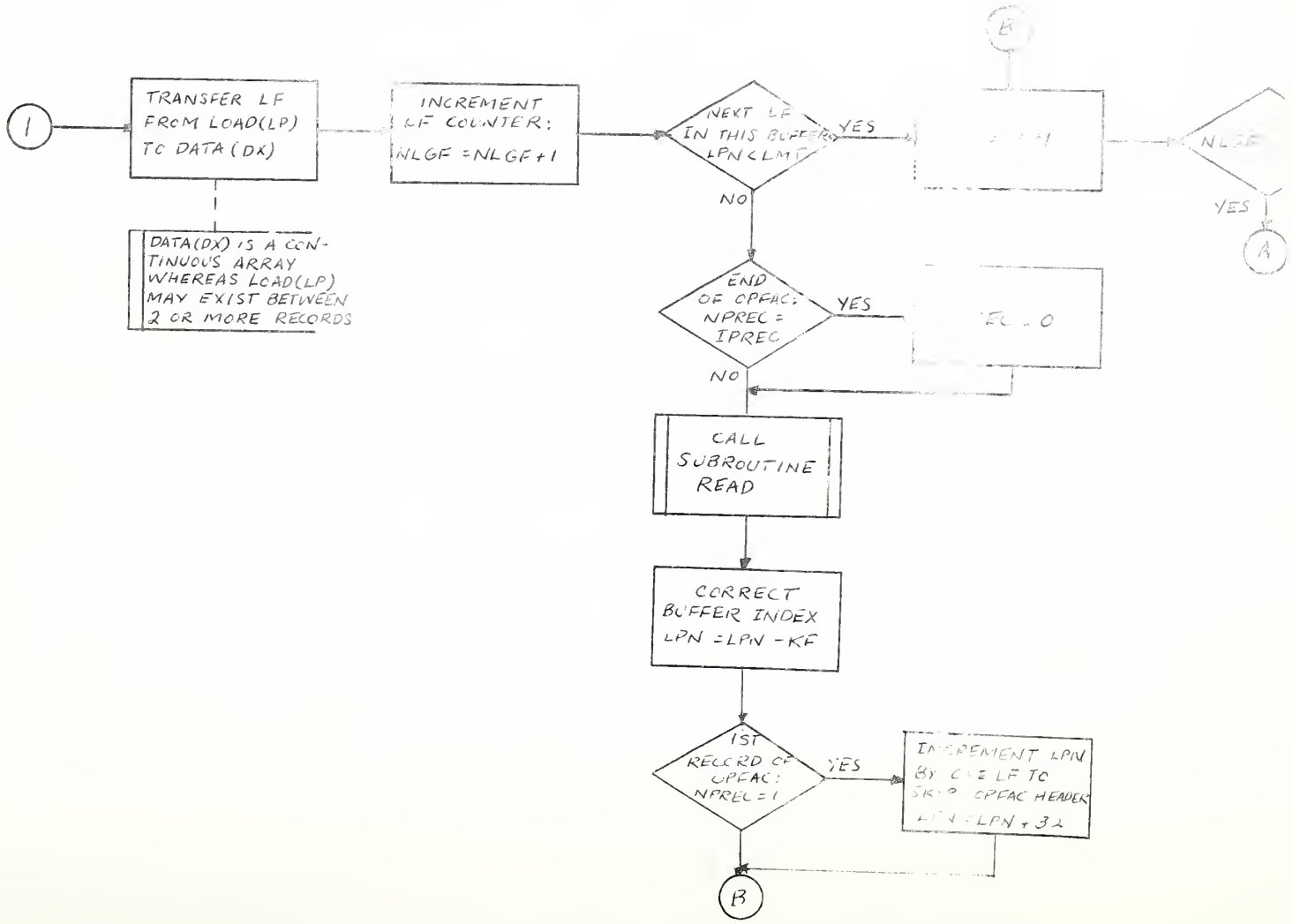

7
NN = N
SUMTOS =MAXTOS
No
GAMMA=100* (TE-MAXTOS)/ SUMTOS-MAXTOS
Yes
GAMMA = 0
M =0 ?
No
NN = N + 1
TOS(NN)=0
DELTA (NN)=0
NEED(NN)=MEED(1)
M=1 ?
NO
YES
Yes
G
NN = 0?
YES
SI=0 AND S2=0?
NO
NEED(1)=0
TOS(1)=0
DELTA(1)=0
No
H
Convert B 13
Write out case
SARSIM CASE FILE
EOF ?
No
H
Yes
stop
MEED(M) =10 ?
Yes
No
MEED(M) =17or NEED(NN) =10,19 ?
YES
NEED(NN) =MEED(M)
No
No
last M ?
Yes
G


SUB ROUTINE NEEDS
D = DI(I,J) / 10
BREAKS OUT FIRST DIGIT OF ASSISTING RESOURCE CODE
D = 6
NO
YES
QUERIES IF ASSISTING RESOURCE IS AN AIRCRAFT
DII(I,J) = 0
YES
1
NO
QUERIES IF AIRCRAFT ASSISTED PERSONNEL
SEE TABLE 2
D = DII(I,J)
NEED(N) = INEED(D)
RET
INEED(D)
DII(I,J) = 0
NO
YES
2
QUERIES ASSISTANCE RENDERED TO PERSONNEL
D12(I,J) = 63, 65, 66, 69, 76
NO
YES
QUERIES PRIMARY ASSISTANCE RENDERED TO PROPERTY
D13(I,J) = 63, 65, 66, 69, 76
NO
YES
3
QUERIES SECONDARY ASSISTANCE RENDERED TO PROPERTY
NEED(N) = 19
NEED(N) = 19 IMPLIES RE AND TOW
RET
3
EXTRACT NEED(N) FROM INEED(D) MATRIX, SEE TABLE 2
D = DII(I,J)
NEED(N) = INEED(D)
INEED(D)
RET

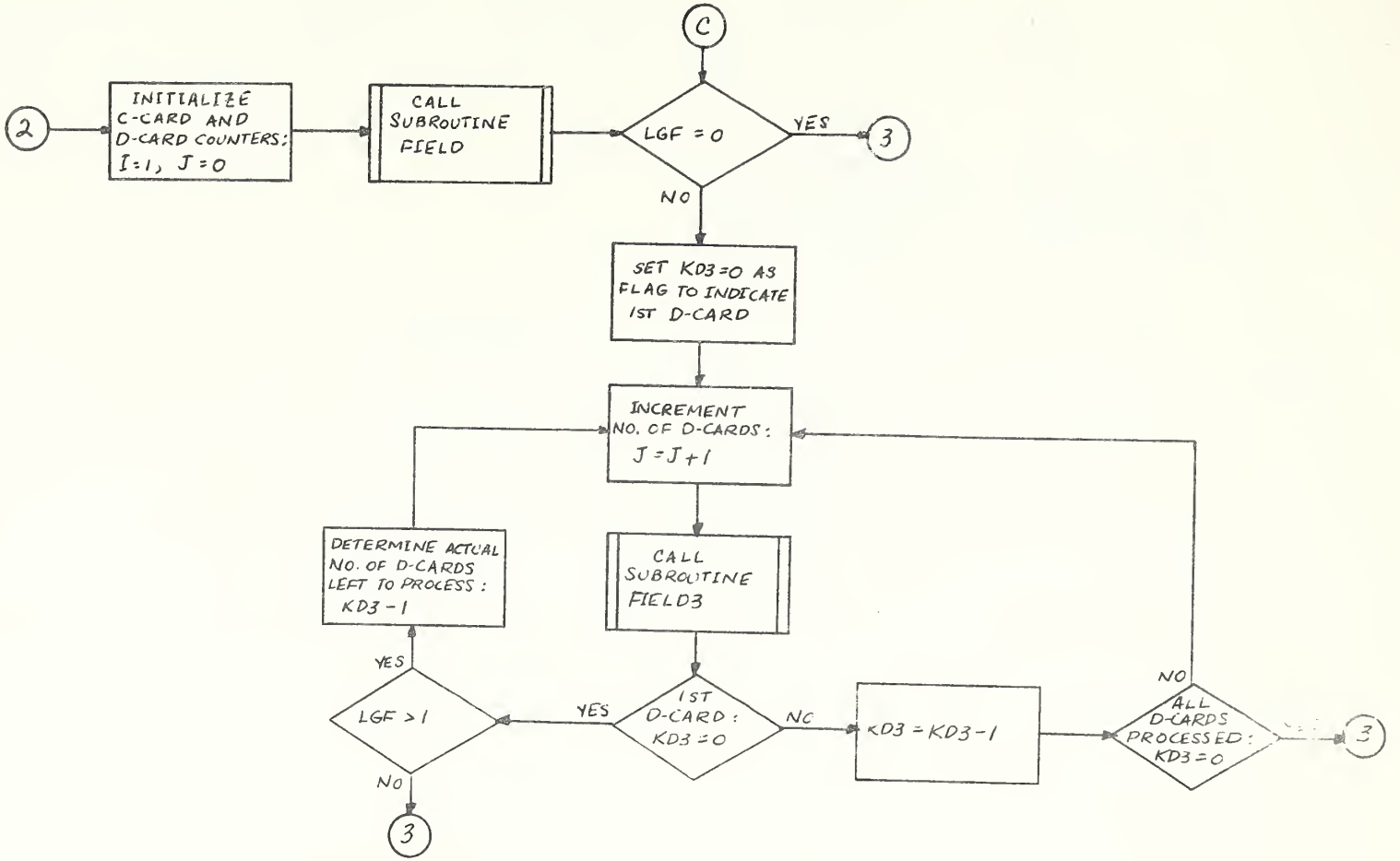

1
18: GENERAL ASSISTANCE RENDERED BY SURFACE OR AIR RESOURCE
D12(I,J) = 0
YES
NEED(N) = 18
RET
NO
D12(I,J) = 63, 65, 66, 69, 76
YES
NEED(N) = INEED(68)
NO
SEE TABLE 2
INEED(D)
B13 ≥ 60, B13 < 70 AND D12(I,J) = 68
NO
YES
NEED(N) = 11
CLIENT AIRCRAFT REQUIRES AIR ESCORT
RET
2
D12 = 0
YES
NEED(N) = 14
RET
14: GENERAL ASSISTANCE RENDERED BY A SURFACE RESOURCE
NO
D13 = 63, 65, 66, 69, 76
YES
NEED(N) = 15
RET
15: TOW; DEWATER & TOW; SURFACE ESCORT; REFLOATED & TOWED
NO
INEED(D)
D = D12(I,J)
NEED(N) = INEED(D)
SEE TABLE 2
D12(I,J) = 68 AND 60 ≤ B13 < 70
YES
NEED(N) = 11
SURFACE RESOURCE STOOD BY BECAUSE IT CANNOT ESCORT AIRCRAFT CLIENT
RET


SUB-ROUTINE NUCASE
TEST FOR ZERO-FILLED LF: LOAD(LP) = 0
No
YES
INITIALIZE LF COUNTER: NLGF = 0
SET LGF = NO. OF LF'S PER CASE
LGF = 0 FOR UNIT CASE WITH NO D-CARDS
SET IPARTS = NO. OF PARTICIPANTS FOR MU CASES
A
CALCULATE BEGINNING OF NEXT LF: LPN = LP + 32
1
CALCULATE BEGINNING OF NEXT LF: LPN = LP + 32
NEXT LF IN THIS BUFFER: LPN < LMT
NO
END OF OPFAC NPREC = IRREC
YES
NPREC = 0
NO
LP = LPN
CALL SUBROUTINE READ
CORRECT BUFFER INDEX: LPN = LPN - KF
1ST RECORD OF OPFAC: NPREC = 1
YES
INCREMENT LPN BY ONE LF TO SKIP OPFAC HEADER LPN = LPN + 32
NO

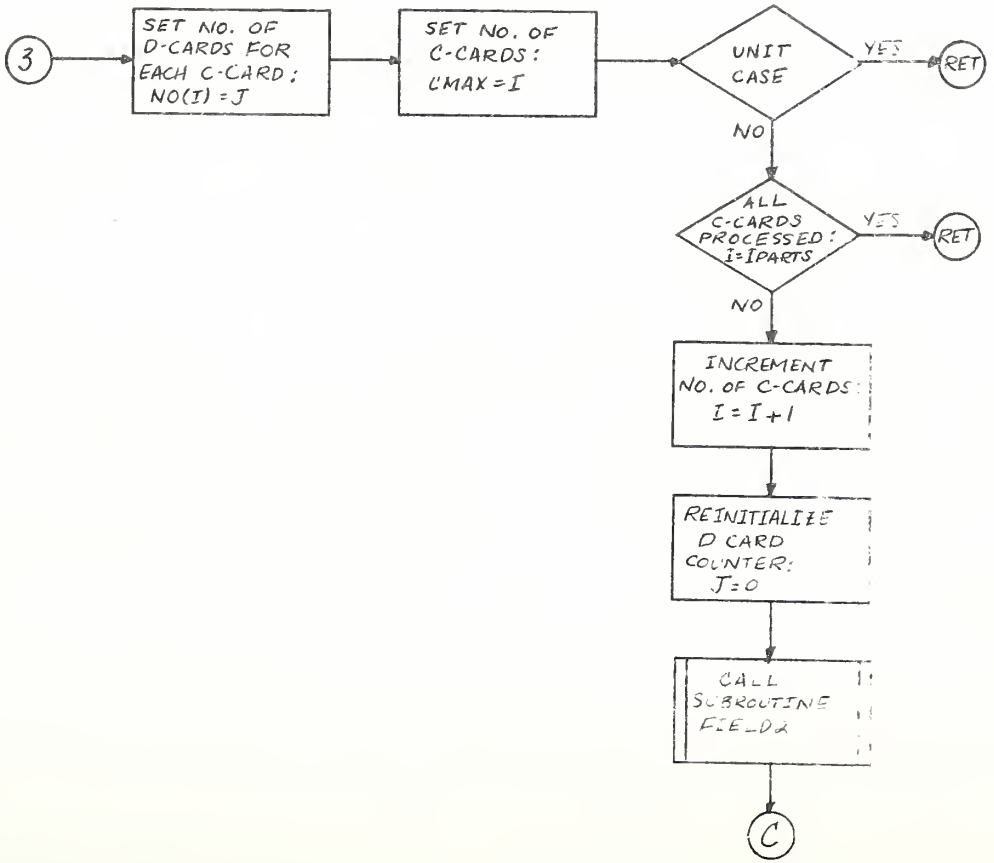
-11-


1
TRANSFER LF FROM LOAD(LP) TO DATA(DX)
DATA(DX) IS A CONTINUOUS ARRAY WHEREAS LOAD(LP) MAY EXIST BETWEEN 2 OR MORE RECORDS
INCREMENT LF COUNTER: NLGF = NLGF + 1
NEXT LF IN THIS BUFFER? LPN < LMT
YES
NO
B
NLGF
YES
A
END OF OPFAC: NPREC = IPREC
YES
NO
CALL SUBROUTINE READ
CORRECT BUFFER INDEX LPN = LPN - KF
1ST RECORD OF OPFAC: NPREC = 1
YES
INCREMENT LPN BY ONE LF TO SKIP OPFAC HEADER
LPN = LPN + 32
B

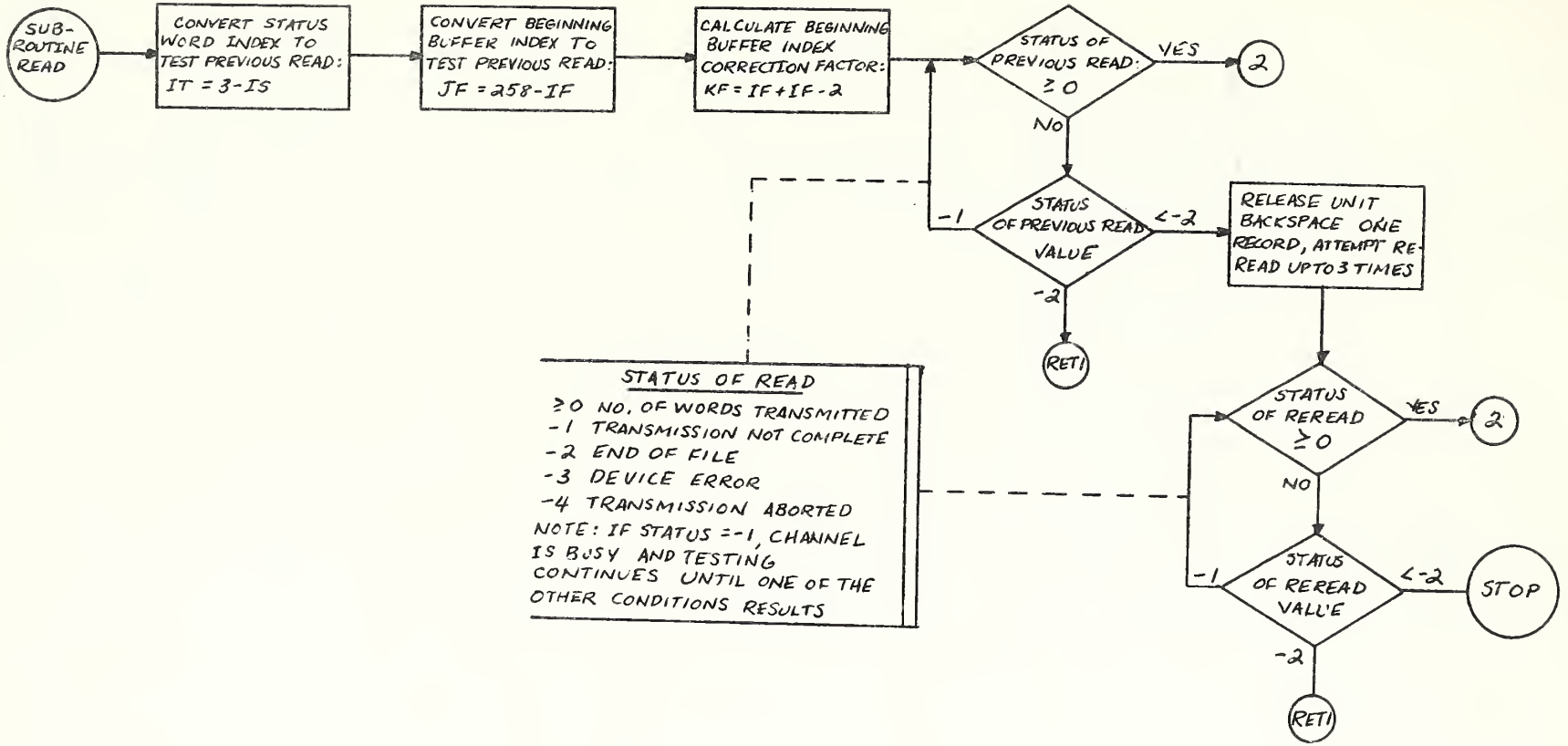
-12-


C
2
INITIALIZE C-CARD AND D-CARD COUNTERS: I=1, J=0
CALL SUBROUTINE FIELD
LGF = 0
YES
3
NO
SET KD3=0 AS FLAG TO INDICATE 1ST D-CARD
INCREMENT NO. OF D-CARDS: J=J+1
CALL SUBROUTINE FIELD3
1ST D-CARD: KD3=0
YES
NO
KD3 = KD3-1
ALL D-CARDS PROCESSED: KD3=0
3
DETERMINE ACTUAL NO. OF D-CARDS LEFT TO PROCESS: KD3-1
LGF > 1
YES
NO
3

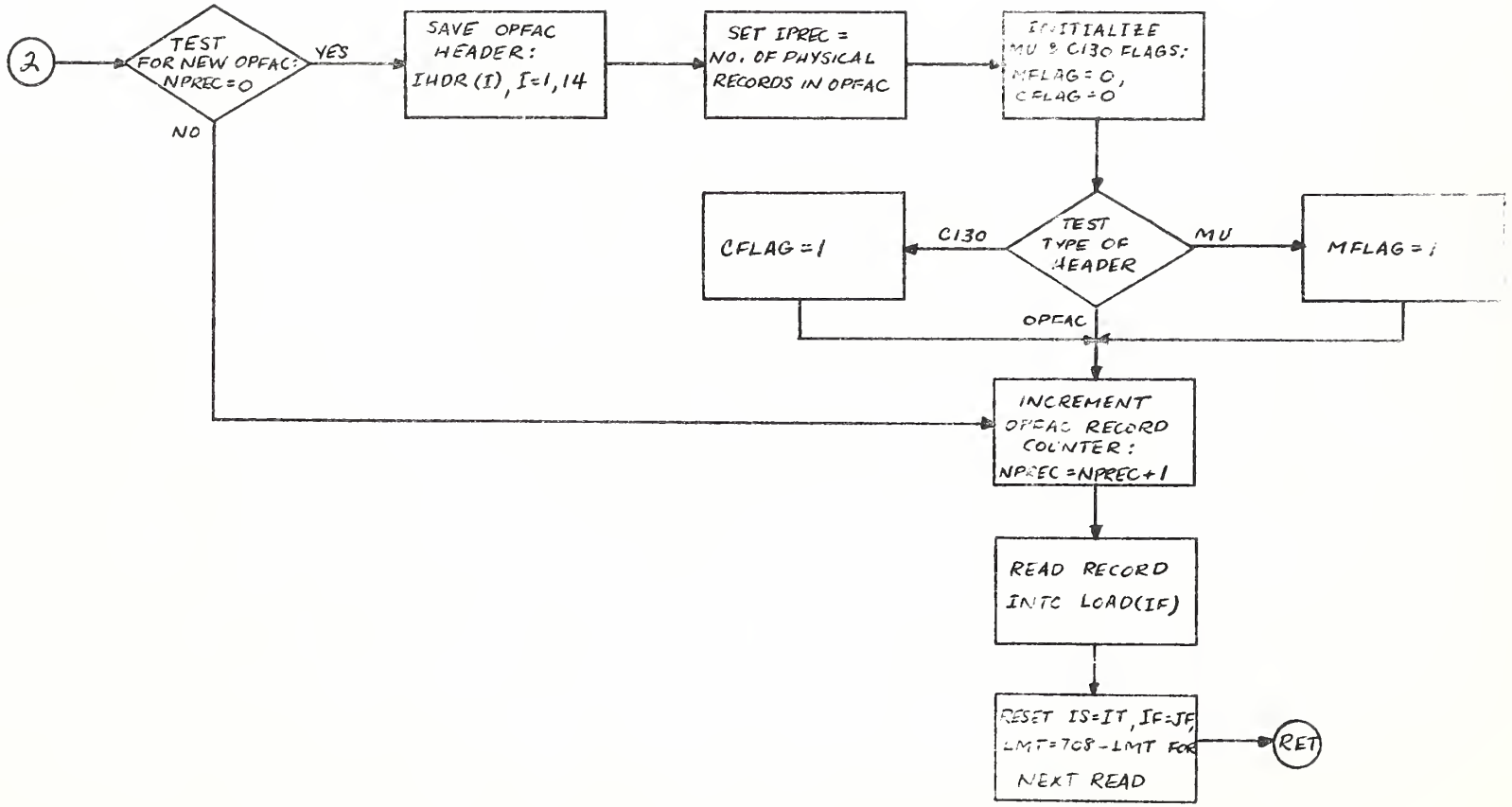
-13-


3
SET NO. OF D-CARDS FOR EACH C-CARD: NO(I) = J
SET NO. OF C-CARDS: CMAX = I
UNIT CASE
YES
RET
NO
ALL C-CARDS PROCESSED: I=IPARTS
YES
RET
NO
INCREMENT NO. OF C-CARDS: I = I + 1
REINITIALIZE D CARD COUNTER: J = 0
CALL SUBROUTINE FIELDA
C

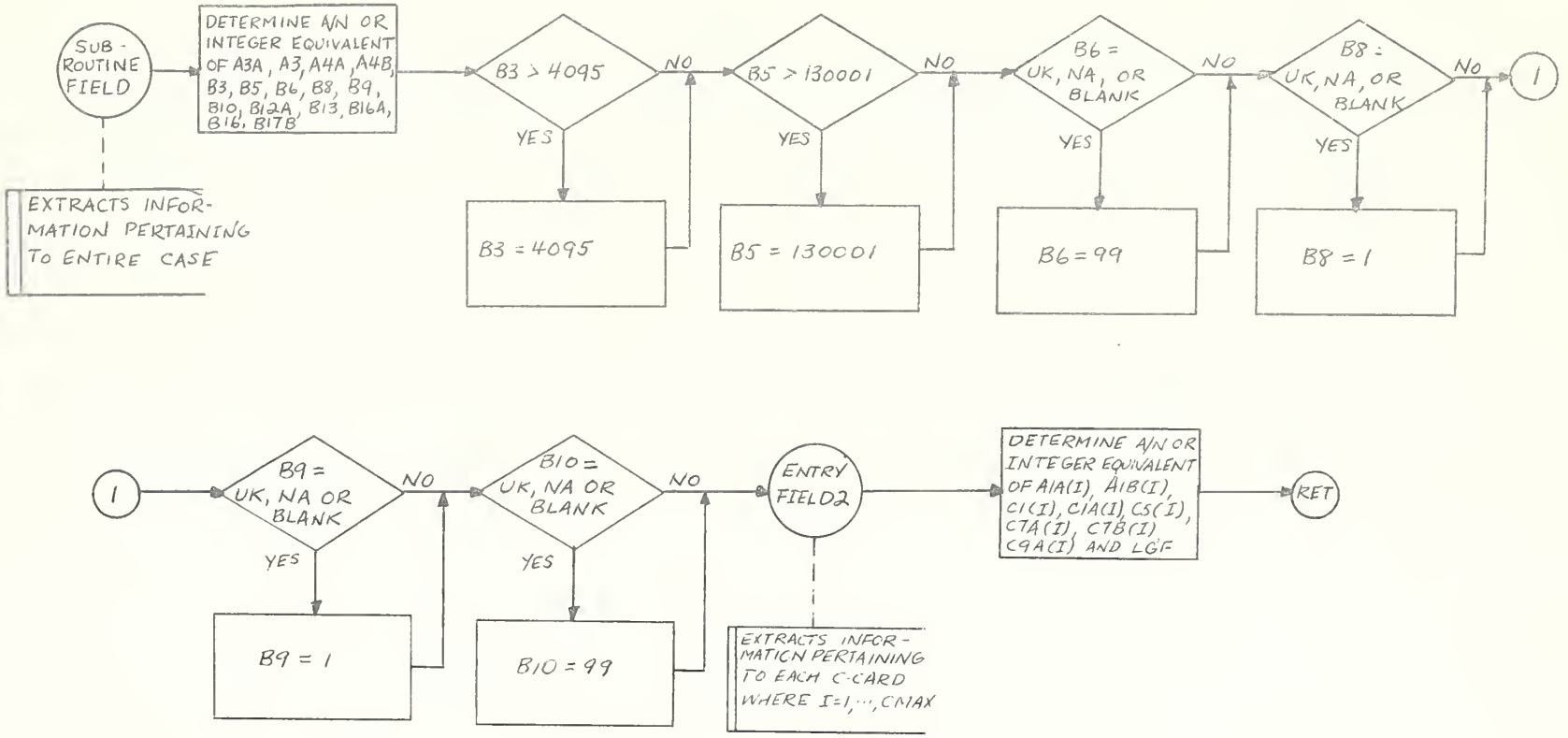
-14-


SUB-ROUTINE READ
CONVERT STATUS WORD INDEX TO TEST PREVIOUS READ: IT = 3-IS
CONVERT BEGINNING BUFFER INDEX TO TEST PREVIOUS READ: JF = 258-IF
CALCULATE BEGINNING BUFFER INDEX CORRECTION FACTOR: KF = IF+IF-2
STATUS OF PREVIOUS READ: ≥ 0
YES
2
No
STATUS OF PREVIOUS READ VALUE
-1
< -2
-2
RETI
RELEASE UNIT BACKSPACE ONE RECORD, ATTEMPT RE-READ UPTO 3 TIMES
STATUS OF READ
≥ 0 NO. OF WORDS TRANSMITTED
-1 TRANSMISSION NOT COMPLETE
-2 END OF FILE
-3 DEVICE ERROR
-4 TRANSMISSION ABORTED
NOTE: IF STATUS = -1, CHANNEL IS BUSY AND TESTING CONTINUES UNTIL ONE OF THE OTHER CONDITIONS RESULTS
STATUS OF REREAD ≥ 0
YES
2
NO
STATUS OF REREAD VALUE
-1
< -2
STOP
-2
RETI

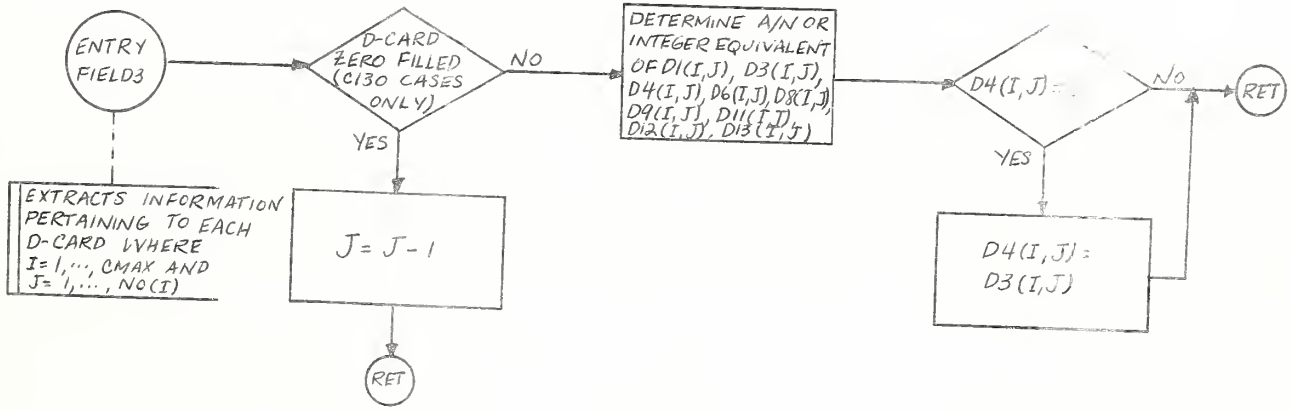
-15-


2
TEST FOR NEW OPFAC: NPREC=0
YES
NO
SAVE OPFAC HEADER: IHDR(I), I=1,14
SET IPREC = NO. OF PHYSICAL RECORDS IN OPFAC
INITIALIZE MU & C130 FLAGS: MFLAG=0, CFLAG=0
TEST TYPE OF HEADER
C130
MU
CFLAG=1
MFLAG=1
OPFAC
INCREMENT OPFAC RECORD COUNTER: NPREC=NPREC+1
READ RECORD INTO LOAD(IF)
RESET IS=IT, IF=JF, LMT=708-LMT FOR NEXT READ
RET

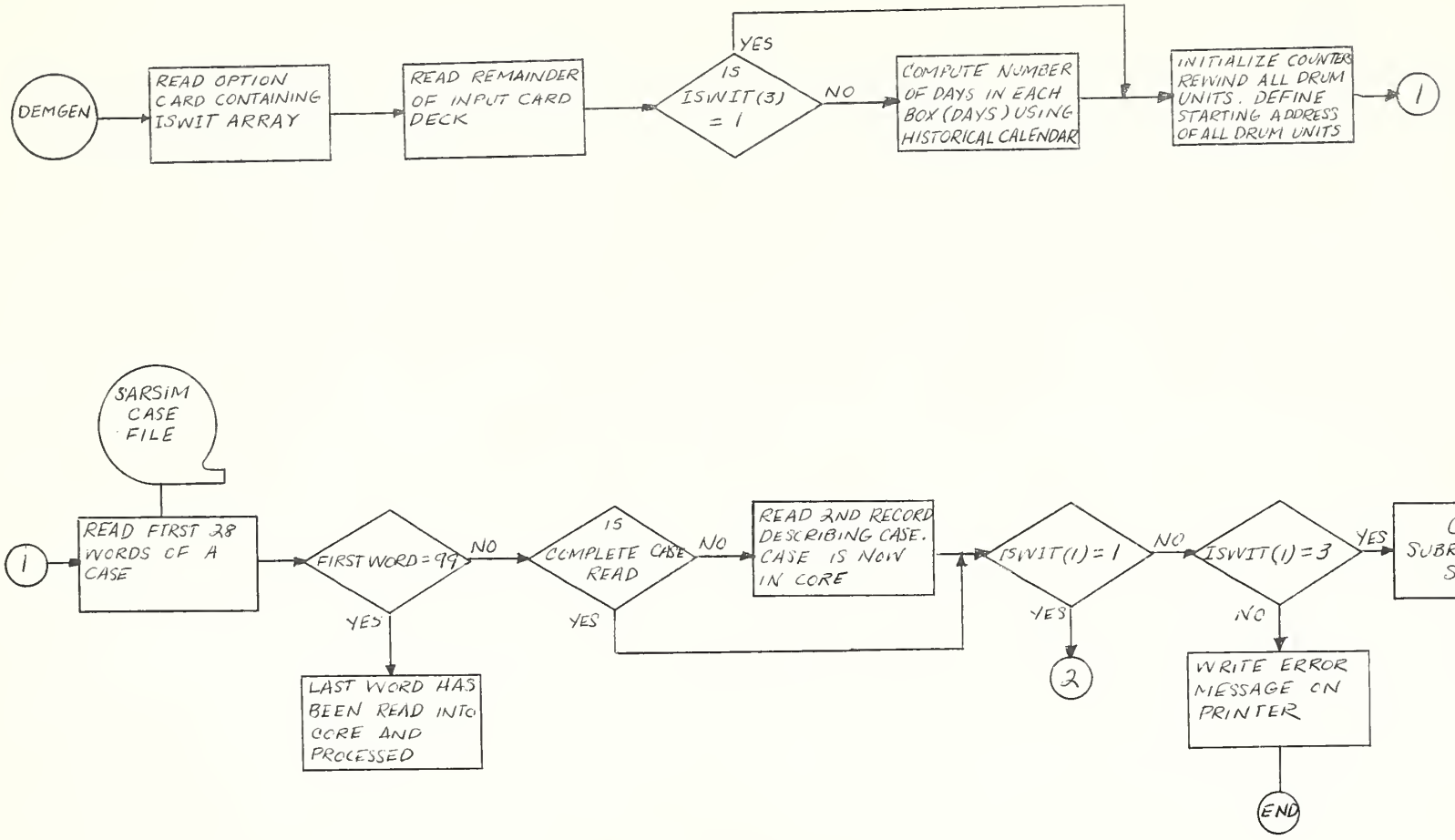
-16-


SUB - ROUTINE FIELD
DETERMINE A/N OR INTEGER EQUIVALENT OF A3A, A3, A4A, A4B, B3, B5, B6, B8, B9, B10, B12A, B13, B16A, B16, B17B
EXTRACTS INFORMATION PERTAINING TO ENTIRE CASE
B3 > 4095
YES
NO
B3 = 4095
B5 > 130001
YES
NO
B5 = 130001
B6 = UK, NA, OR BLANK
YES
NO
B6 = 99
B8 = UK, NA, OR BLANK
YES
NO
B8 = 1
1
1
B9 = UK, NA OR BLANK
YES
NO
B9 = 1
B10 = UK, NA OR BLANK
YES
NO
B10 = 99
ENTRY FIELD2
EXTRACTS INFORMATION PERTAINING TO EACH C-CARD WHERE I=1,···,CMAX
DETERMINE A/N OR INTEGER EQUIVALENT OF A1A(I), A1B(I), C1(I), C1A(I), C5(I), C7A(I), C7B(I) C9A(I) AND LGF
RET

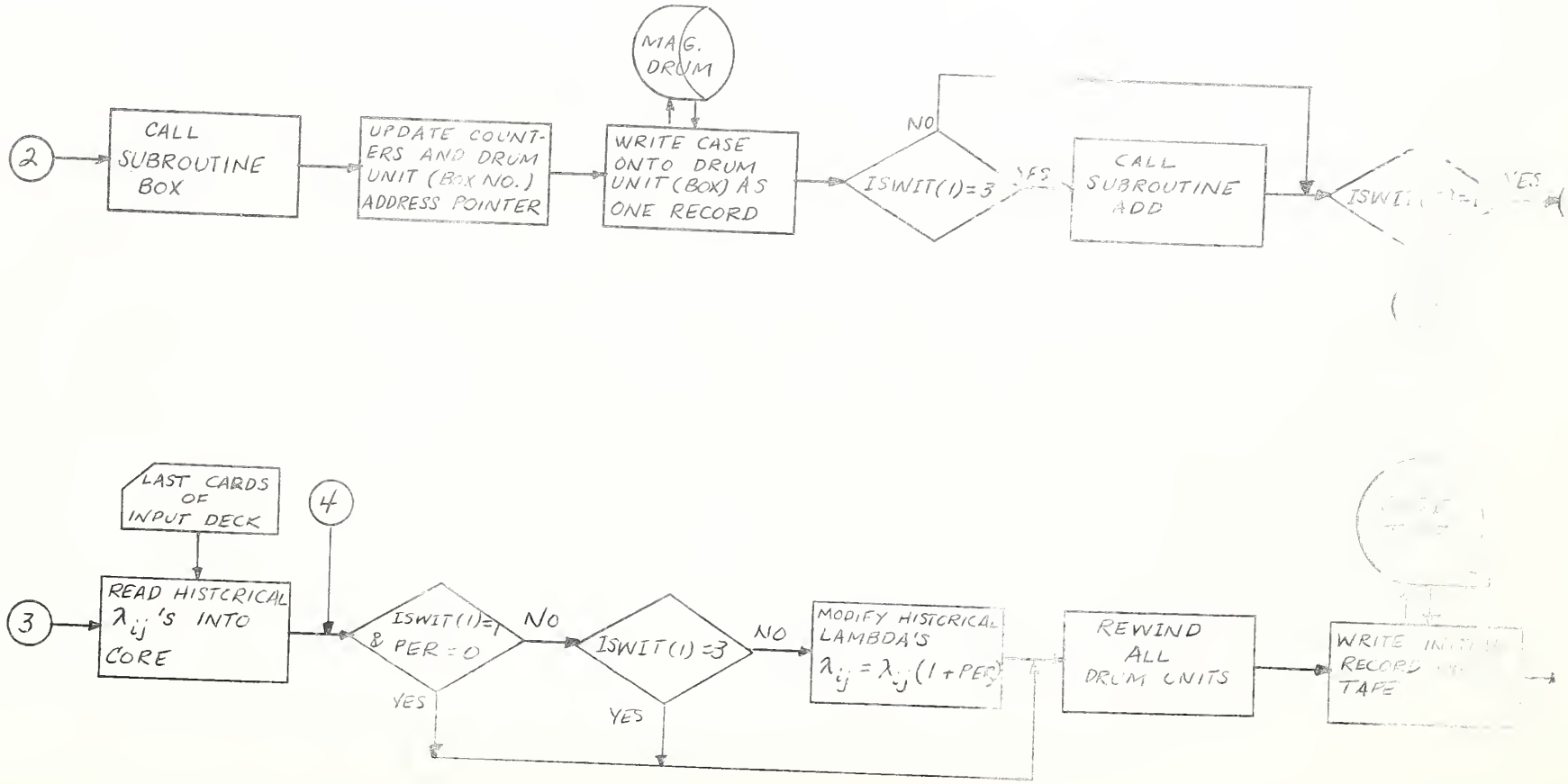
-17-


ENTRY FIELD3
D-CARD ZERO FILLED (C130 CASES ONLY)
NO
YES
DETERMINE A/N OR INTEGER EQUIVALENT OF D1(I,J), D3(I,J), D4(I,J), D6(I,J), D8(I,J), D9(I,J), D11(I,J), D12(I,J), D13(I,J)
D4(I,J) =
NO
YES
RET
EXTRACTS INFORMATION PERTAINING TO EACH D-CARD WHERE I=1,..., CMAX AND J=1,..., NO(I)
J = J - 1
RET
D4(I,J) = D3(I,J)

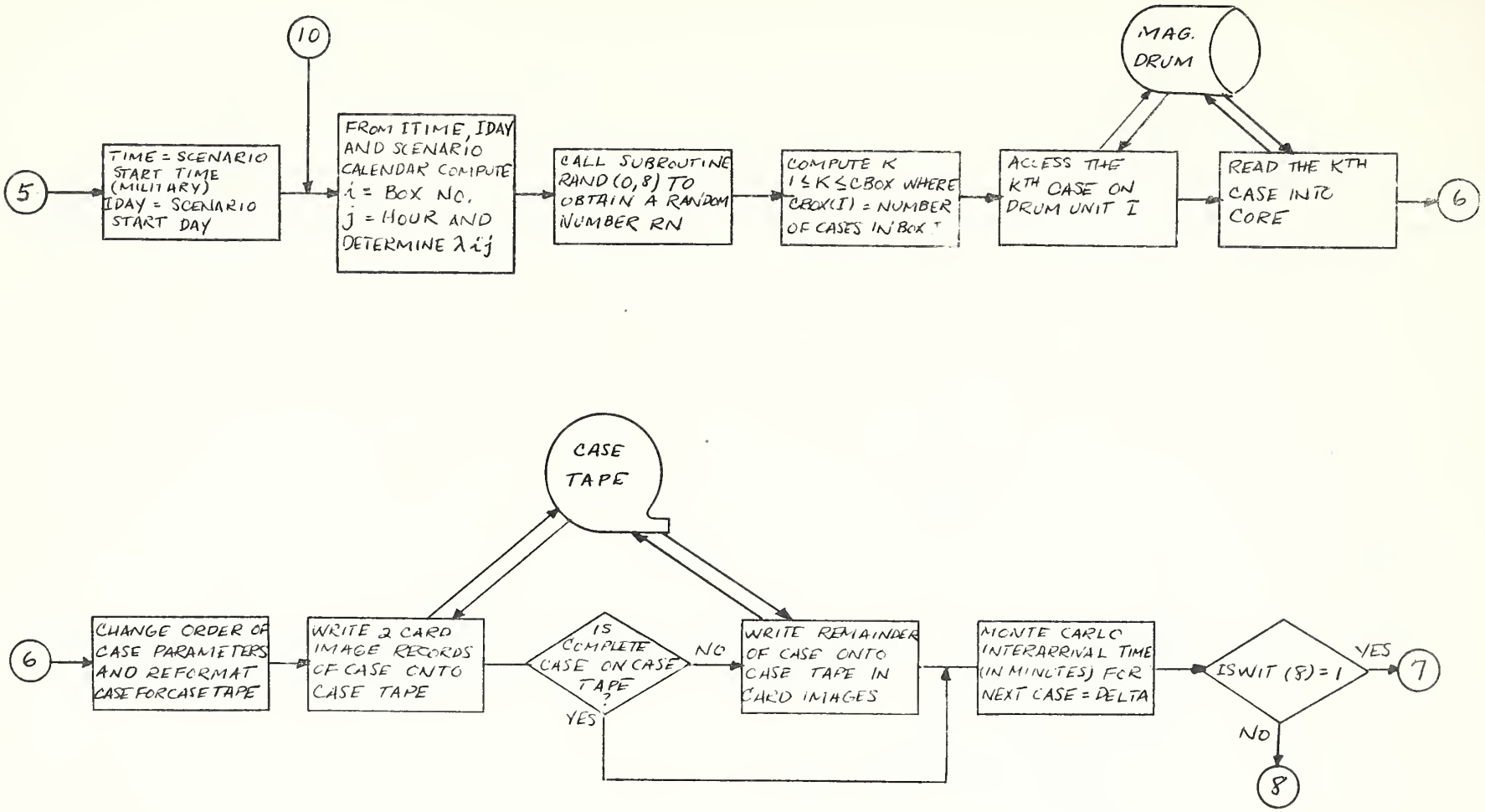
-18-


DEMGEN
READ OPTION CARD CONTAINING ISWIT ARRAY
READ REMAINDER OF INPUT CARD DECK
IS ISWIT(3) = 1
YES
NO
COMPUTE NUMBER OF DAYS IN EACH BOX (DAYS) USING HISTORICAL CALENDAR
INITIALIZE COUNTERS REWIND ALL DRUM UNITS. DEFINE STARTING ADDRESS OF ALL DRUM UNITS
1
SARSIM CASE FILE
1
READ FIRST 28 WORDS OF A CASE
FIRST WORD = 99
NO
YES
LAST WORD HAS BEEN READ INTO CORE AND PROCESSED
IS COMPLETE CASE READ
NO
YES
READ 2ND RECORD DESCRIBING CASE. CASE IS NOW IN CORE
ISWIT(1) = 1
NO
YES
2
ISWIT(1) = 3
YES
NO
WRITE ERROR MESSAGE ON PRINTER
END
CALL SUBROUTINE SELECT
2

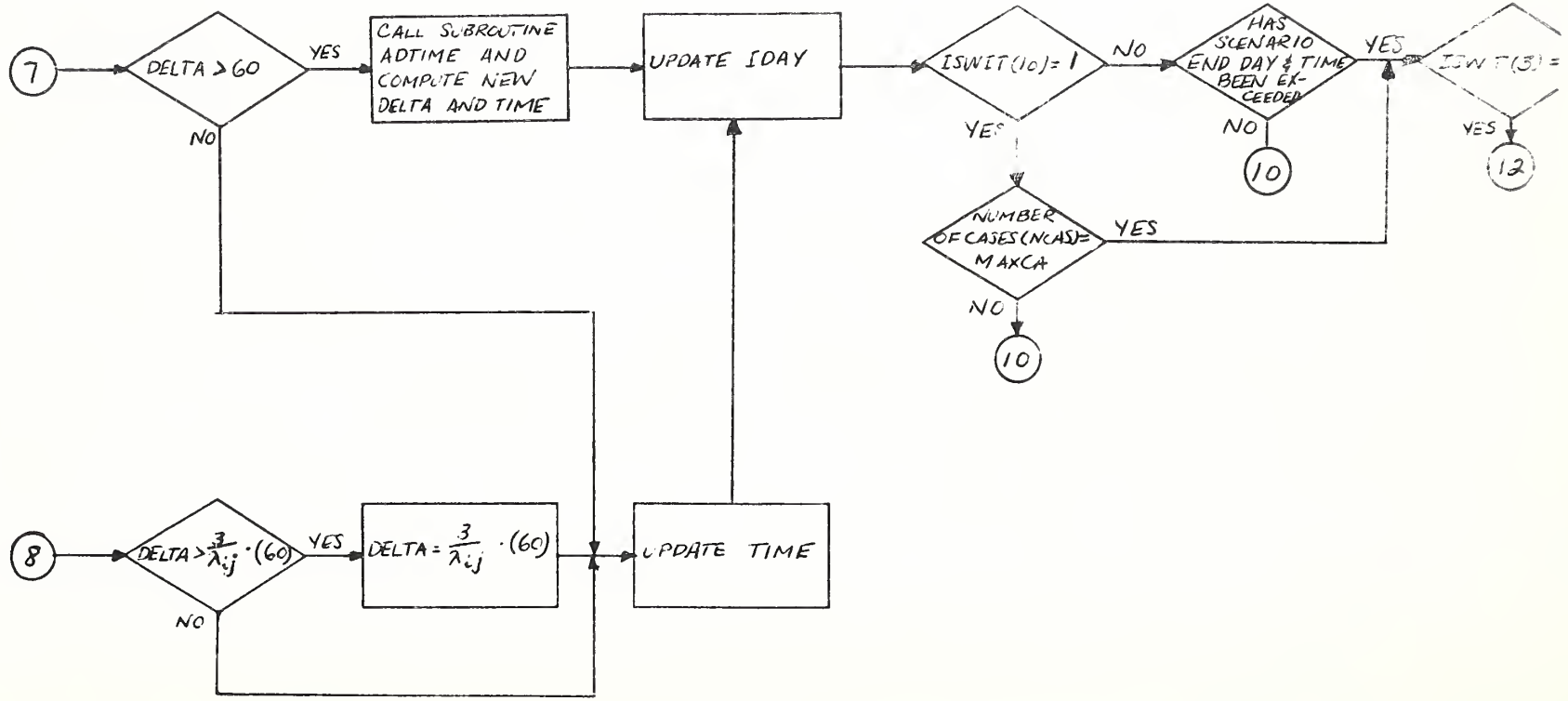
-19-


2
CALL SUBROUTINE BOX
UPDATE COUNTERS AND DRUM UNIT (BOX NO.) ADDRESS POINTER
MAG. DRUM
WRITE CASE ONTO DRUM UNIT (BOX) AS ONE RECORD
NO
ISWIT(1)=3
YES
CALL SUBROUTINE ADD
YES
LAST CARDS OF INPUT DECK
4
3
READ HISTORICAL λ_{ij}'S INTO CORE
ISWIT(1)=1 & PER=0
YES
NO
ISWIT(1)=3
YES
NO
MODIFY HISTORICAL LAMBDA'S
$\lambda_{ij} = \lambda_{ij}(1+PER)$
REWIND ALL DRUM UNITS
WRITE
RECORD
TAPE

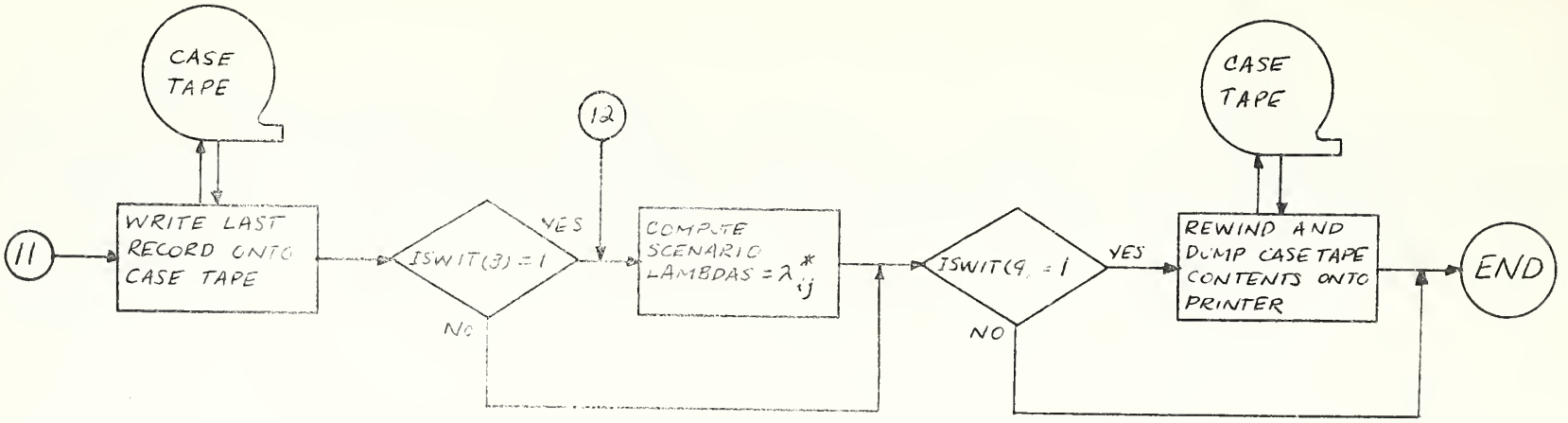
-20-

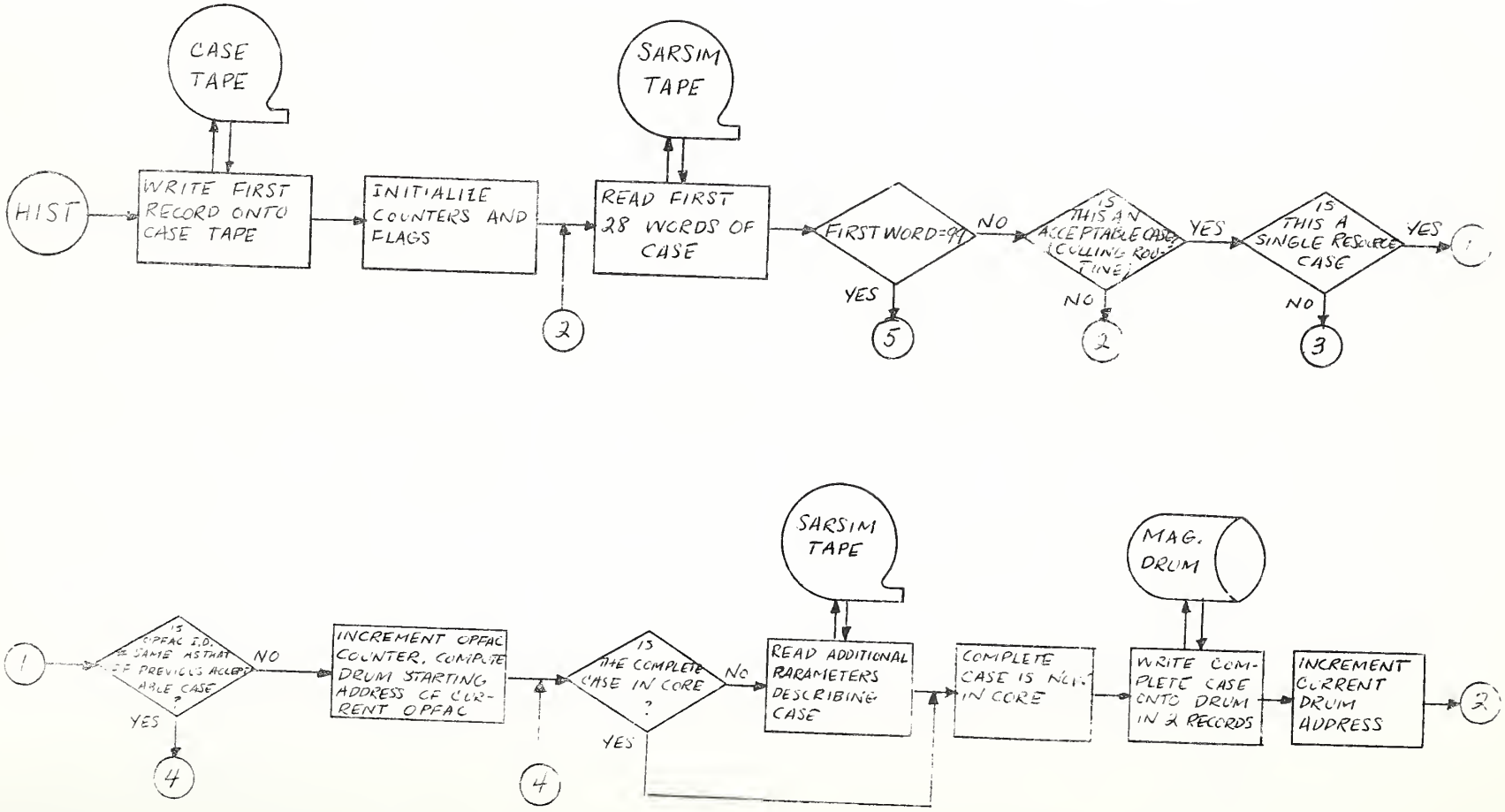

5
TIME = SCENARIO START TIME (MILITARY)
IDAY = SCENARIO START DAY
10
FROM ITIME, IDAY AND SCENARIO CALENDAR COMPUTE
i = BOX NO.
j = HOUR AND DETERMINE λij
CALL SUBROUTINE RAND (0,8) TO OBTAIN A RANDOM NUMBER RN
COMPUTE K
1 ≤ K ≤ CBOX WHERE
CBOX(I) = NUMBER OF CASES IN BOX I
MAG. DRUM
ACCESS THE KTH CASE ON DRUM UNIT I
READ THE KTH CASE INTO CORE
6
CASE TAPE
6
CHANGE ORDER OF CASE PARAMETERS AND REFORMAT CASE FOR CASE TAPE
WRITE 2 CARD IMAGE RECORDS OF CASE ONTO CASE TAPE
IS COMPLETE CASE ON CASE TAPE ?
NO
YES
WRITE REMAINDER OF CASE ONTO CASE TAPE IN CARD IMAGES
MONTE CARLO INTERARRIVAL TIME (IN MINUTES) FOR NEXT CASE = DELTA
ISWIT (8) = 1
YES
7
NO
8

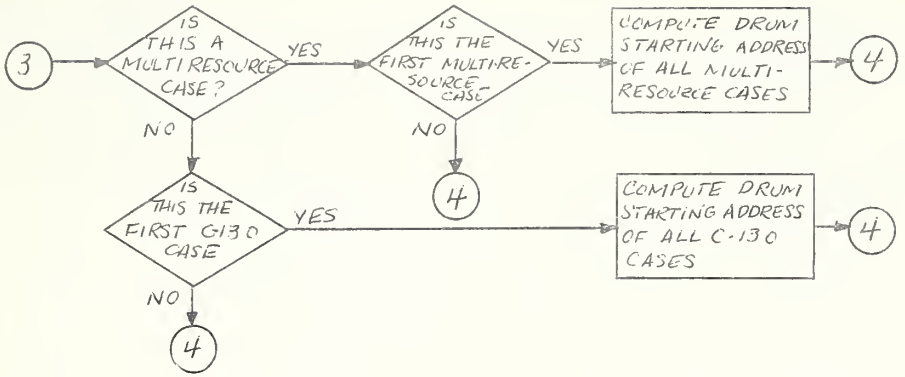
-21-

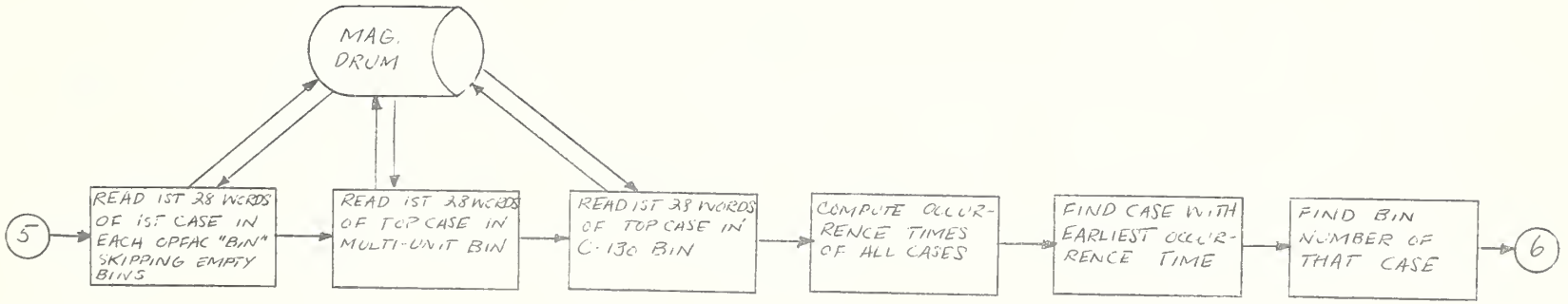

7
DELTA > 60
YES
NO
CALL SUBROUTINE ADTIME AND COMPUTE NEW DELTA AND TIME
UPDATE IDAY
ISWIT(10) = 1
YES
NO
HAS SCENARIO END DAY & TIME BEEN EXCEEDED
YES
NO
10
ISWIT(3) =
YES
12
NUMBER OF CASES (NCAS) = MAXCA
YES
NO
10
8
DELTA > 3/λij · (60)
YES
NO
DELTA = 3/λij · (60)
UPDATE TIME

-22-

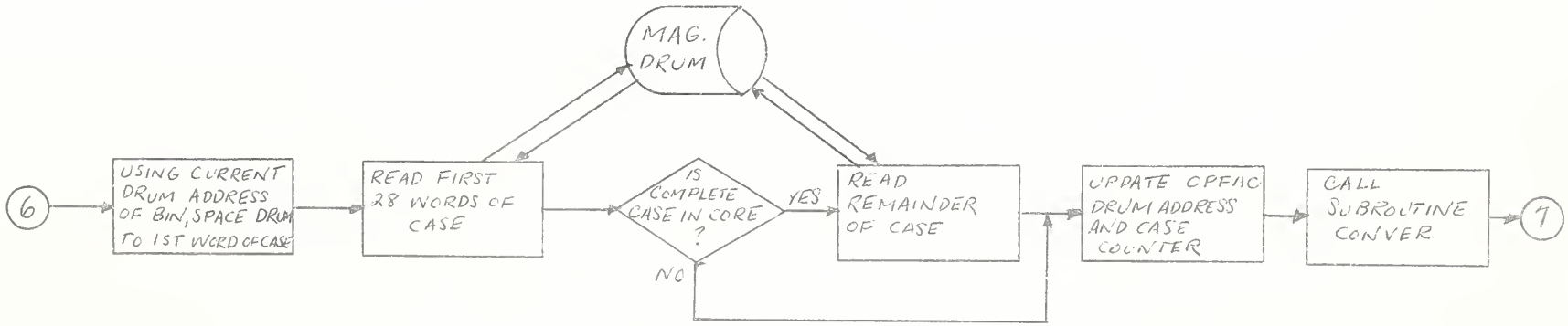

CASE TAPE
11
WRITE LAST RECORD ONTO CASE TAPE
ISWIT(3) = 1
YES
NO
12
COMPUTE SCENARIO LAMBDAS = λ^*_{ij}
ISWIT(9) = 1
YES
NO
CASE TAPE
REWIND AND DUMP CASE TAPE CONTENTS ONTO PRINTER
END

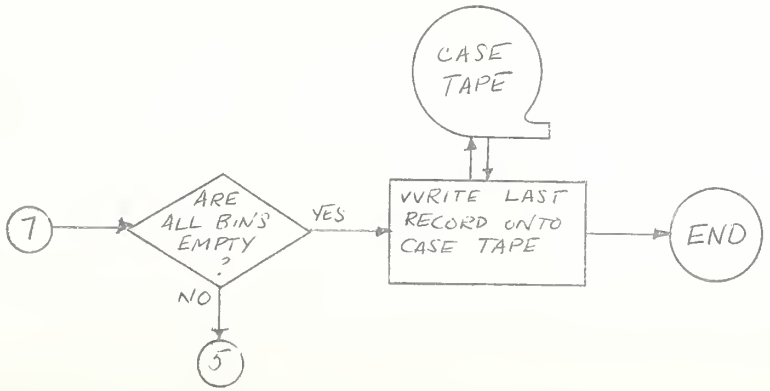

CASE TAPE
SARSIM TAPE
HIST
WRITE FIRST RECORD ONTO CASE TAPE
INITIALIZE COUNTERS AND FLAGS
READ FIRST 28 WORDS OF CASE
FIRST WORD=9?
NO
YES
IS THIS AN ACCEPTABLE CASE? (CULLING ROUTINE)
YES
NO
IS THIS A SINGLE RESOURCE CASE
YES
NO
1
2
3
4
5
SARSIM TAPE
MAG. DRUM
IS OPFAC I.D. SAME AS THAT OF PREVIOUS ACCEPTABLE CASE?
NO
YES
INCREMENT OPFAC COUNTER, COMPUTE DRUM STARTING ADDRESS OF CURRENT OPFAC
IS THE COMPLETE CASE IN CORE?
NO
YES
READ ADDITIONAL PARAMETERS DESCRIBING CASE
COMPLETE CASE IS NOW IN CORE
WRITE COMPLETE CASE ONTO DRUM IN 2 RECORDS
INCREMENT CURRENT DRUM ADDRESS


3
IS THIS A MULTI RESOURCE CASE?
YES
NO
IS THIS THE FIRST MULTI-RESOURCE CASE
YES
NO
COMPUTE DRUM STARTING ADDRESS OF ALL MULTI-RESOURCE CASES
4
4
IS THIS THE FIRST C-130 CASE
YES
NO
COMPUTE DRUM STARTING ADDRESS OF ALL C-130 CASES
4
4


MAG. DRUM
5
READ 1ST 28 WORDS OF 1ST CASE IN EACH OPFAC "BIN" SKIPPING EMPTY BINS
READ 1ST 28 WORDS OF TOP CASE IN MULTI-UNIT BIN
READ 1ST 28 WORDS OF TOP CASE IN C-130 BIN
COMPUTE OCCURRENCE TIMES OF ALL CASES
FIND CASE WITH EARLIEST OCCURRENCE TIME
FIND BIN NUMBER OF THAT CASE
6

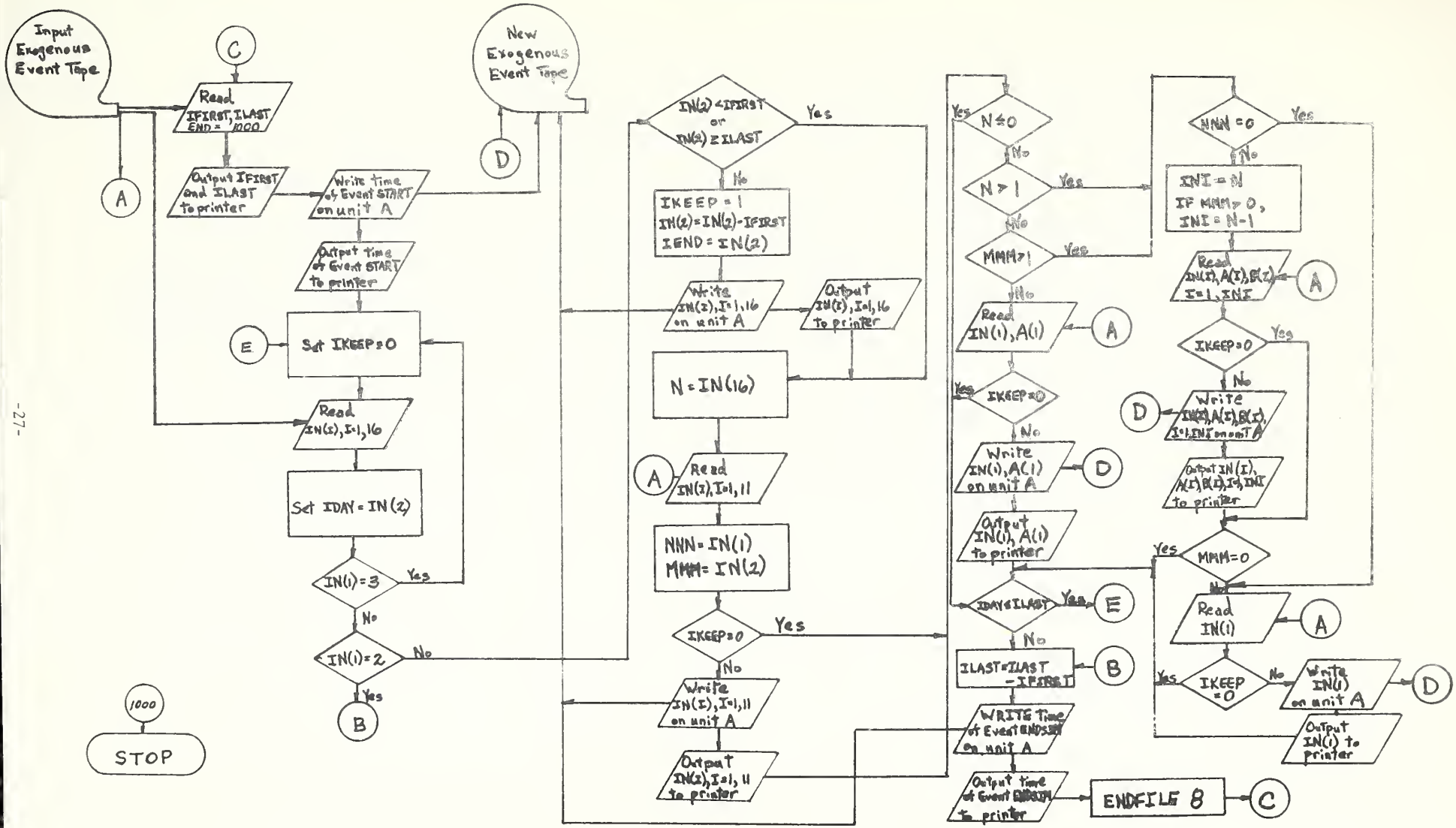
-25-


MAG. DRUM
6
USING CURRENT DRUM ADDRESS OF BIN, SPACE DRUM TO 1ST WORD OF CASE
READ FIRST 28 WORDS OF CASE
IS COMPLETE CASE IN CORE ?
YES
NO
READ REMAINDER OF CASE
UPDATE OPFIC DRUM ADDRESS AND CASE COUNTER
CALL SUBROUTINE CONVER
7


CASE TAPE
7
ARE ALL BIN'S EMPTY ?
YES
NO
WRITE LAST RECORD ONTO CASE TAPE
END
5

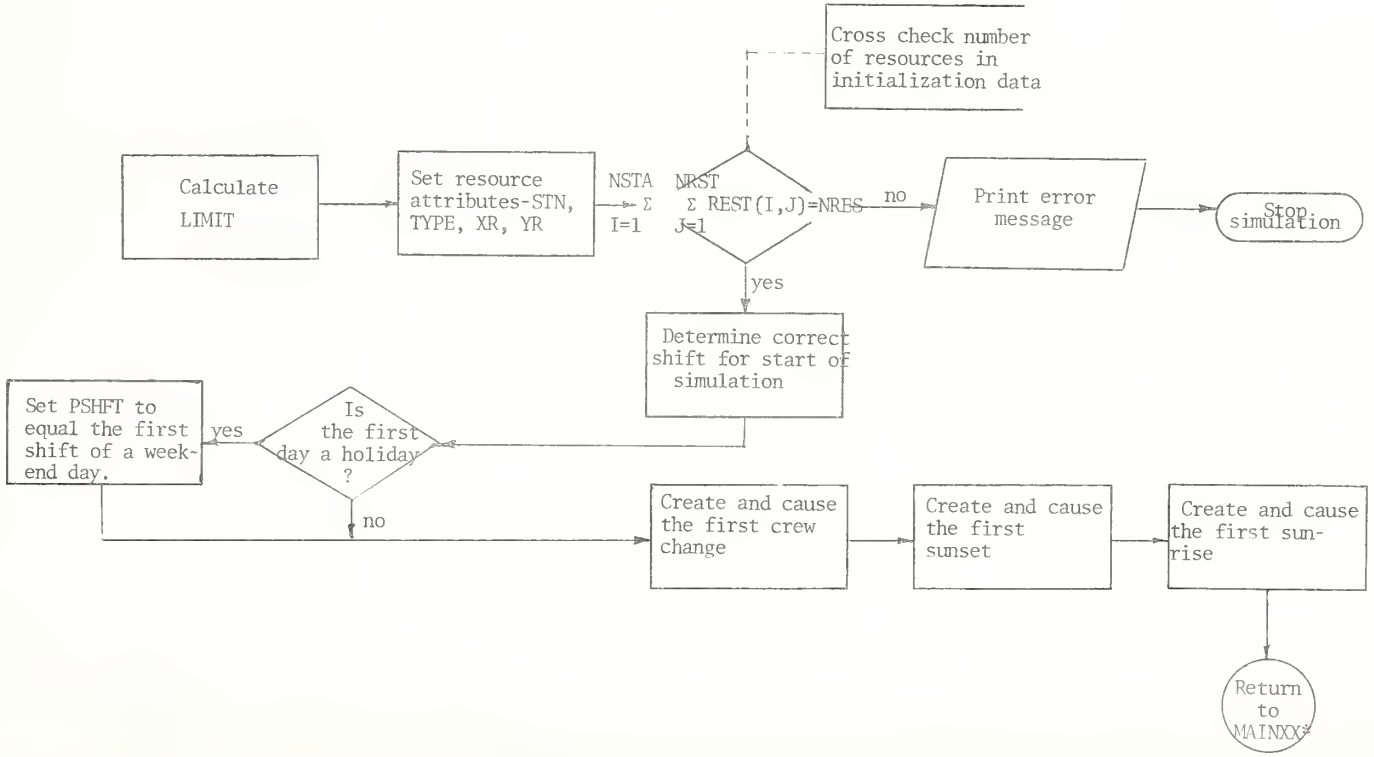
-26-

PROGRAM SELECT



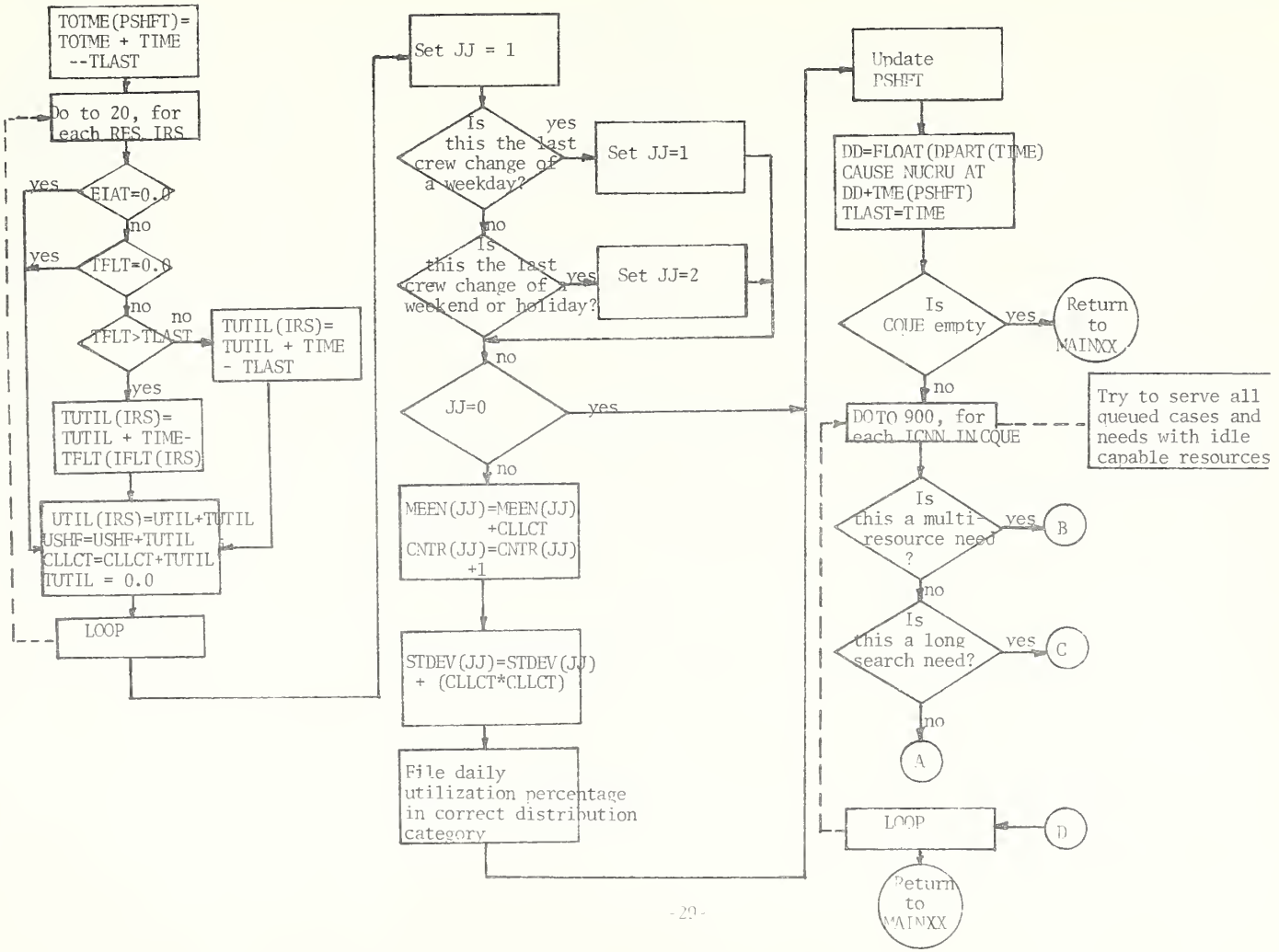
-27-

EXOGENOUS EVENT START

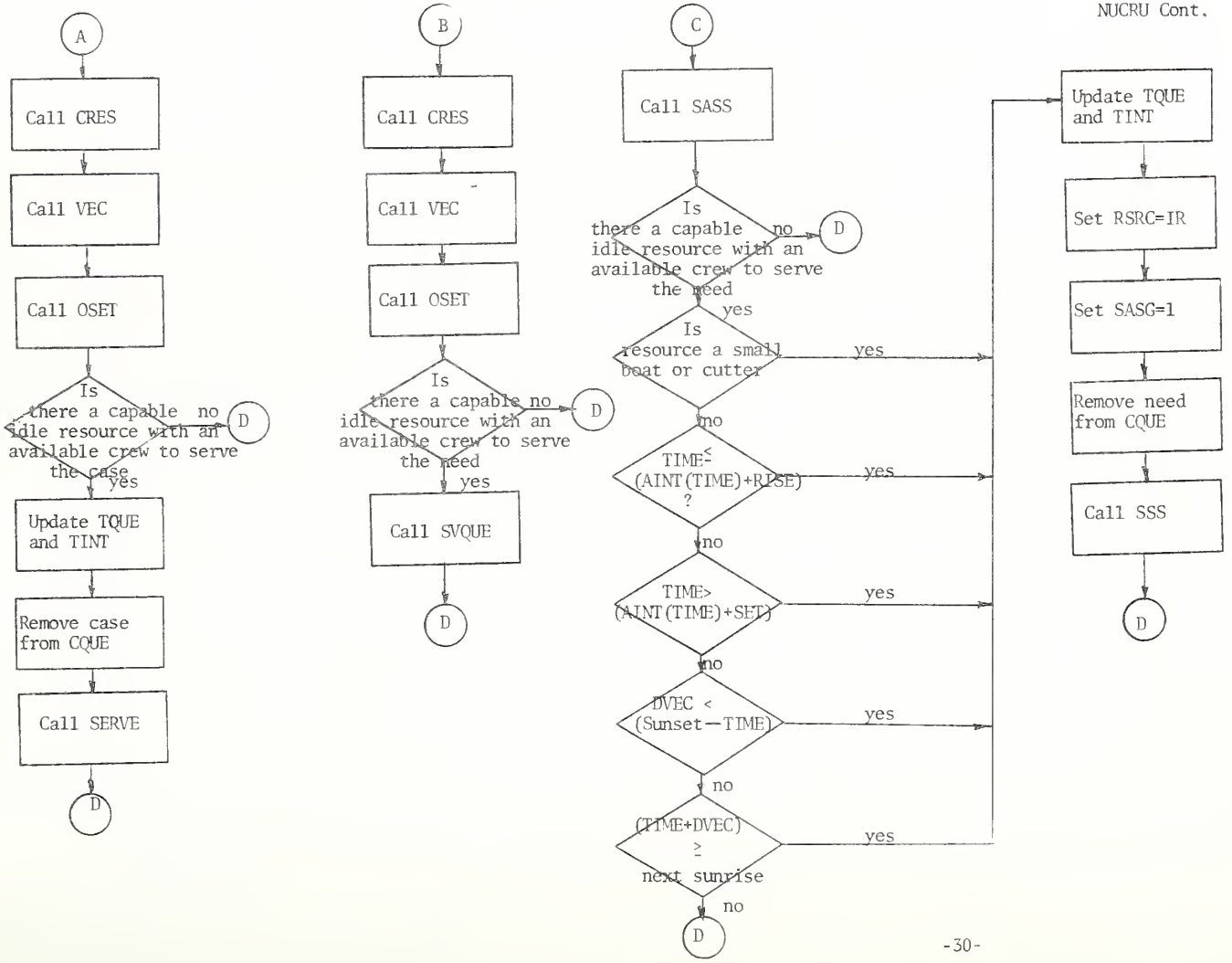

Cross check number of resources in initialization data
Calculate LIMIT
Set resource attributes-STN, TYPE, XR, YR
NSTA Σ I=1
NRST Σ REST(I,J)=NRES J=1
no
Print error message
Stop simulation
yes
Determine correct shift for start of simulation
Set PSHFT to equal the first shift of a week-end day.
yes
Is the first day a holiday ?
no
Create and cause the first crew change
Create and cause the first sunset
Create and cause the first sun-rise
Return to MAINXX*

*MAINXX is the timing routine

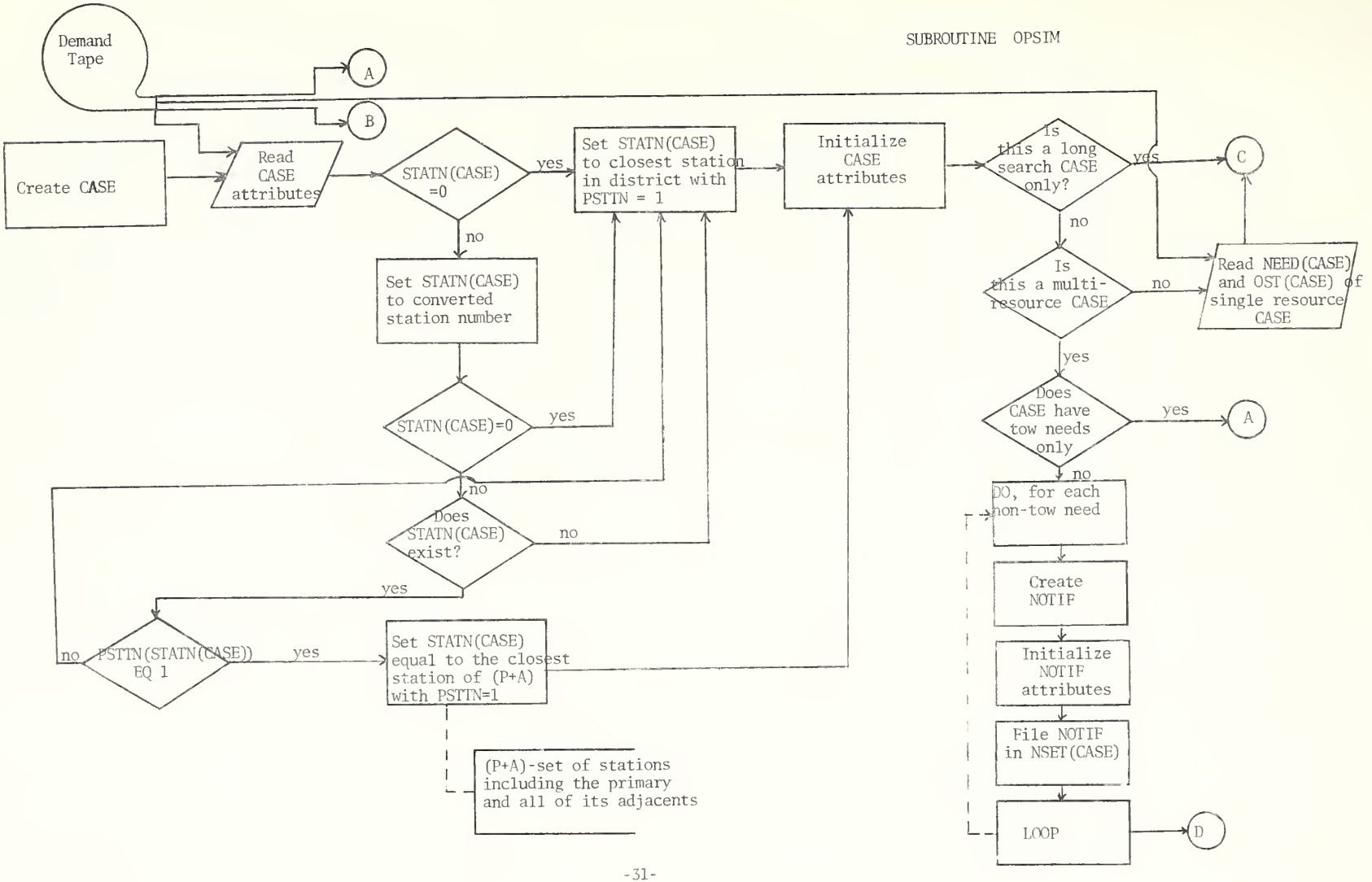
-28-


TOTME(PSHFT)=
TOTME + TIME
--TLAST
Do to 20, for
each RES, IRS
yes
EIAT=0.0
no
yes
IFLT=0.0
no
no
IFLT>TLAST
TUTIL(IRS)=
TUTIL + TIME
- TLAST
yes
TUTIL(IRS)=
TUTIL + TIME-
TFLT(IFLT(IRS))
UTIL(IRS)=UTIL+TUTIL
USHF=USHF+TUTIL
CLLCT=CLLCT+TUTIL
TUTIL = 0.0
LOOP
Set JJ = 1
Is
this the last
crew change of
a weekday?
yes
Set JJ=1
no
Is
this the last
crew change of
weekend or holiday?
yes
Set JJ=2
no
JJ=0
yes
no
MEEN(JJ)=MEEN(JJ)
+CLLCT
CNTR(JJ)=CNTR(JJ)
+1
STDEV(JJ)=STDEV(JJ)
+ (CLLCT*CLLCT)
File daily
utilization percentage
in correct distribution
category
Update
PSHFT
DD=FLOAT(DPART(TIME)
CAUSE NUCRU AT
DD+TME(PSHFT)
TLAST=TIME
Is
COUE empty
yes
Return
to
MAINXX
no
DOTO 900, for
each ICNN IN COUE
Try to serve all
queued cases and
needs with idle
capable resources
Is
this a multi-
resource need
?
yes
B
no
Is
this a long
search need?
yes
C
no
A
LOOP
D
Return
to
MAINXX

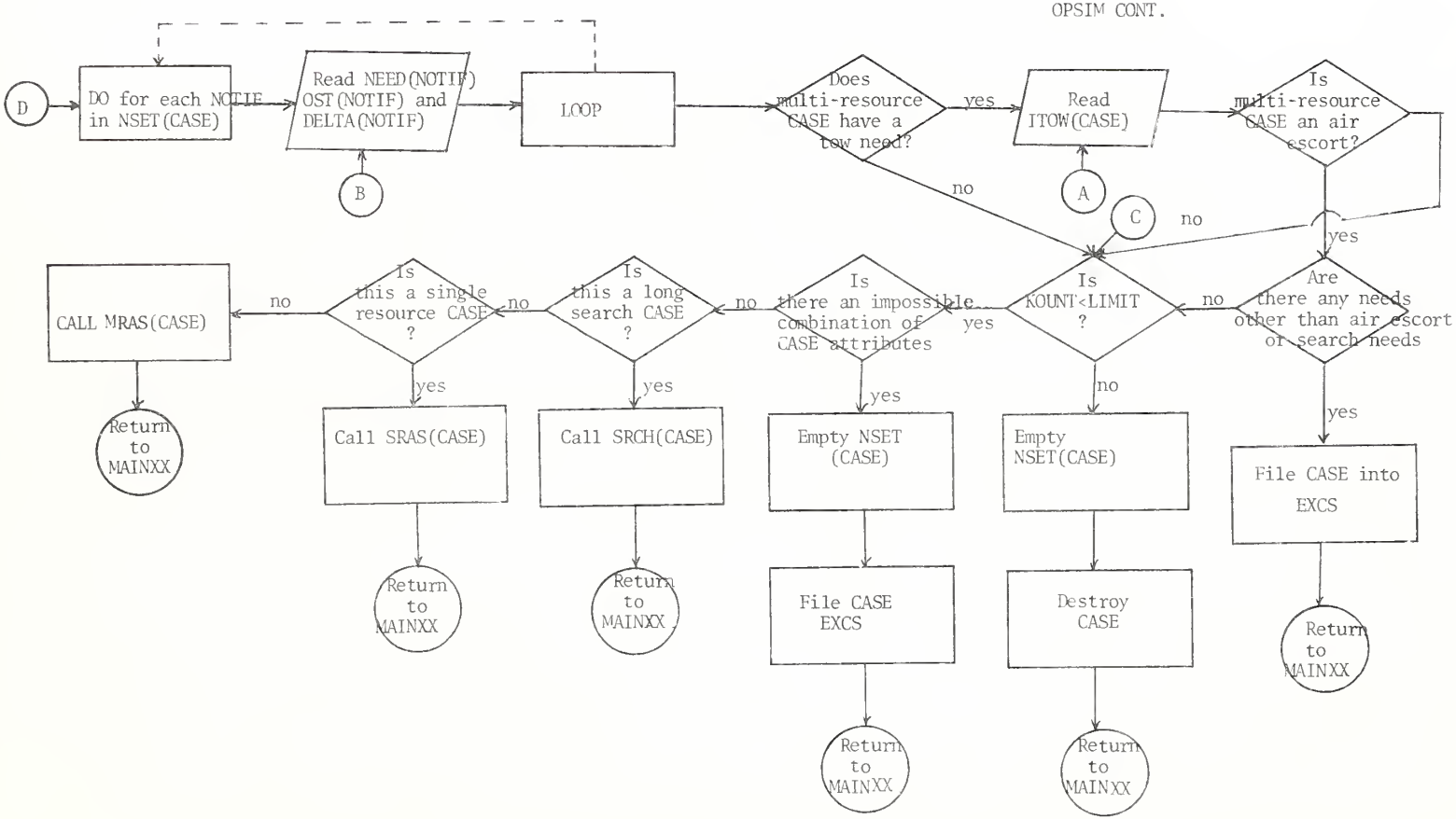
-29-


A
Call CRES
Call VEC
Call OSET
Is there a capable idle resource with an available crew to serve the case
no
D
yes
Update TQUE and TINT
Remove case from CQUE
Call SERVE
D
B
Call CRES
Call VEC
Call OSET
Is there a capable idle resource with an available crew to serve the need
no
D
yes
Call SVQUE
D
C
Call SASS
Is there a capable idle resource with an available crew to serve the need
no
D
yes
Is resource a small boat or cutter
yes
no
TIME≤ (AINT(TIME)+RISE) ?
yes
no
TIME> (AINT(TIME)+SET)
yes
no
DVEC < (Sunset−TIME)
yes
no
(TIME+DVEC) ≥ next sunrise
yes
no
D
Update TQUE and TINT
Set RSRC=IR
Set SASG=1
Remove need from CQUE
Call SSS
D

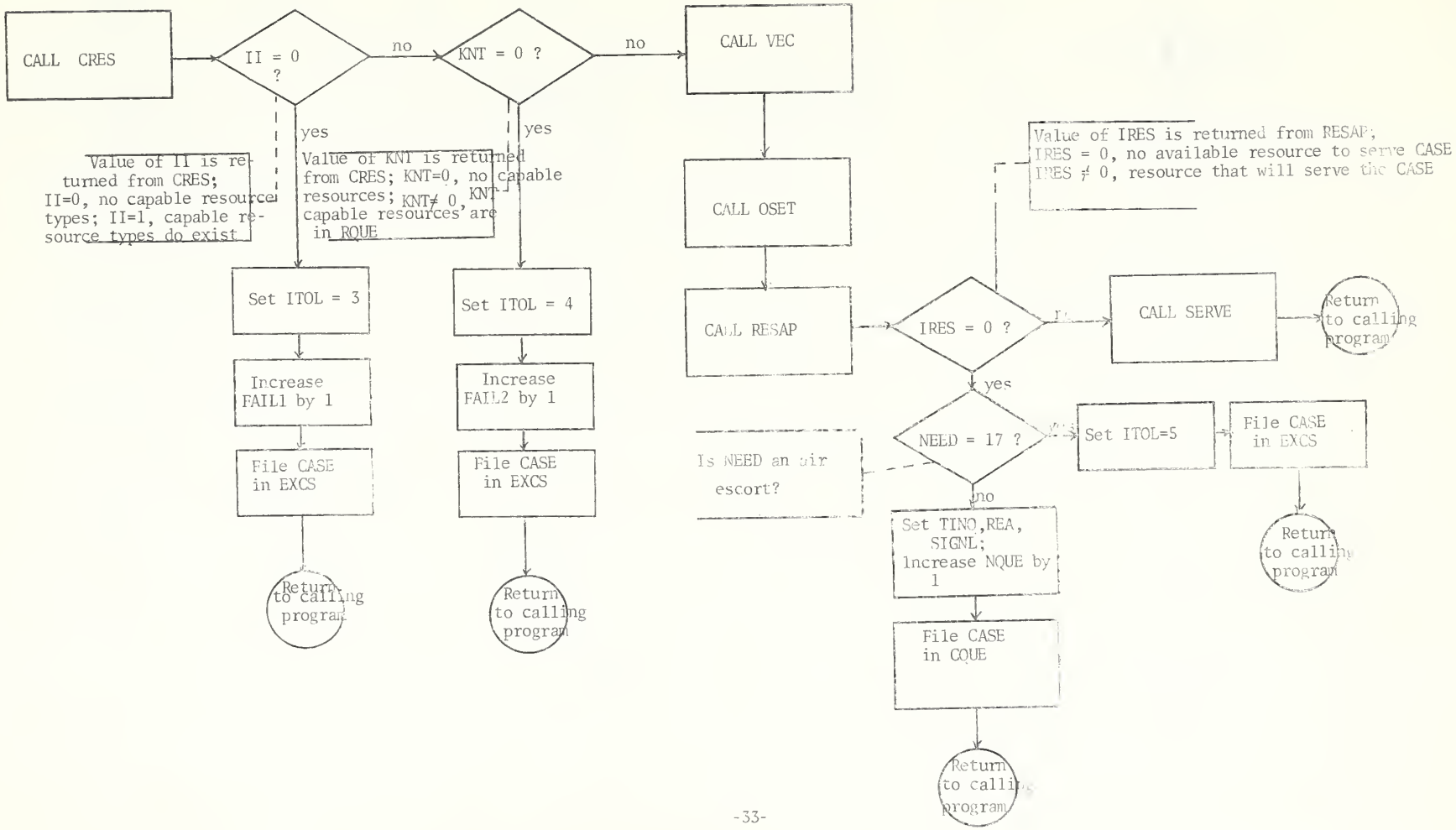
-30-


Demand Tape
A
B
Create CASE
Read CASE attributes
STATN(CASE) =0
yes
no
Set STATN(CASE) to closest station in district with PSTTN = 1
Initialize CASE attributes
Is this a long search CASE only?
yes
C
no
Is this a multi-resource CASE
no
Read NEED(CASE) and OST(CASE) of single resource CASE
yes
Does CASE have tow needs only
yes
A
no
DO, for each non-tow need
Create NOTIF
Initialize NOTIF attributes
File NOTIF in NSET(CASE)
LOOP
D
Set STATN(CASE) to converted station number
STATN(CASE)=0
yes
no
Does STATN(CASE) exist?
no
yes
no
PSTTN(STATN(CASE)) EQ 1
yes
Set STATN(CASE) equal to the closest station of (P+A) with PSTTN=1
(P+A)-set of stations including the primary and all of its adjacents

-31-

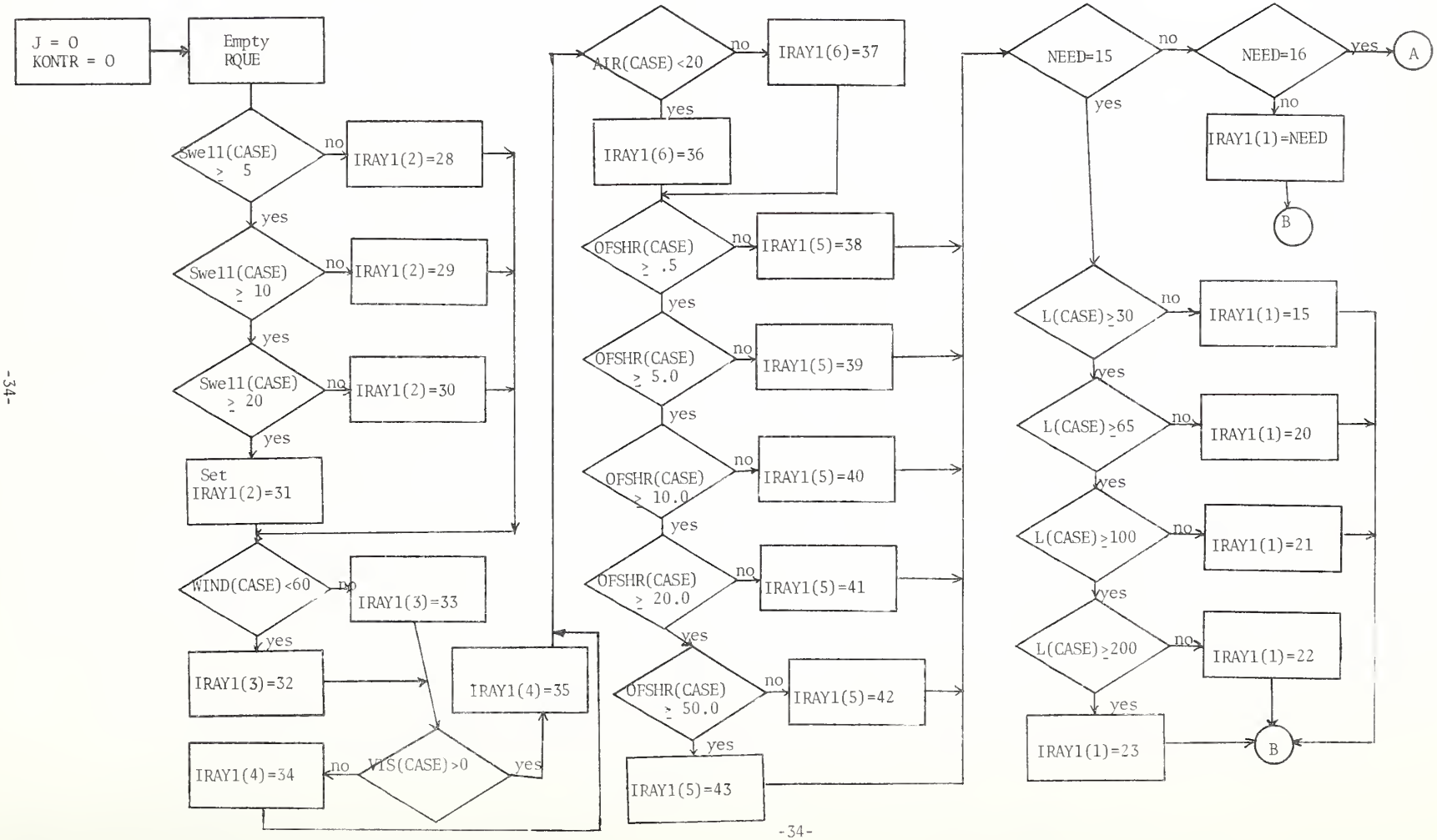

D
DO for each NOTIF in NSET(CASE)
Read NEED(NOTIF) OST(NOTIF) and DELTA(NOTIF)
B
LOOP
Does multi-resource CASE have a tow need?
yes
no
Read ITOW(CASE)
A
Is multi-resource CASE an air escort?
C
no
yes
Are there any needs other than air escort or search needs
no
Is KOUNT<LIMIT ?
no
yes
Is there an impossible combination of CASE attributes
yes
Is this a long search CASE ?
no
yes
Is this a single resource CASE ?
no
yes
CALL MRAS(CASE)
no
Return to MAINXX
Call SRAS(CASE)
Return to MAINXX
Call SRCH(CASE)
Return to MAINXX
Empty NSET (CASE)
File CASE EXCS
Return to MAINXX
Empty NSET(CASE)
Destroy CASE
Return to MAINXX
yes
File CASE into EXCS
Return to MAINXX

-32-

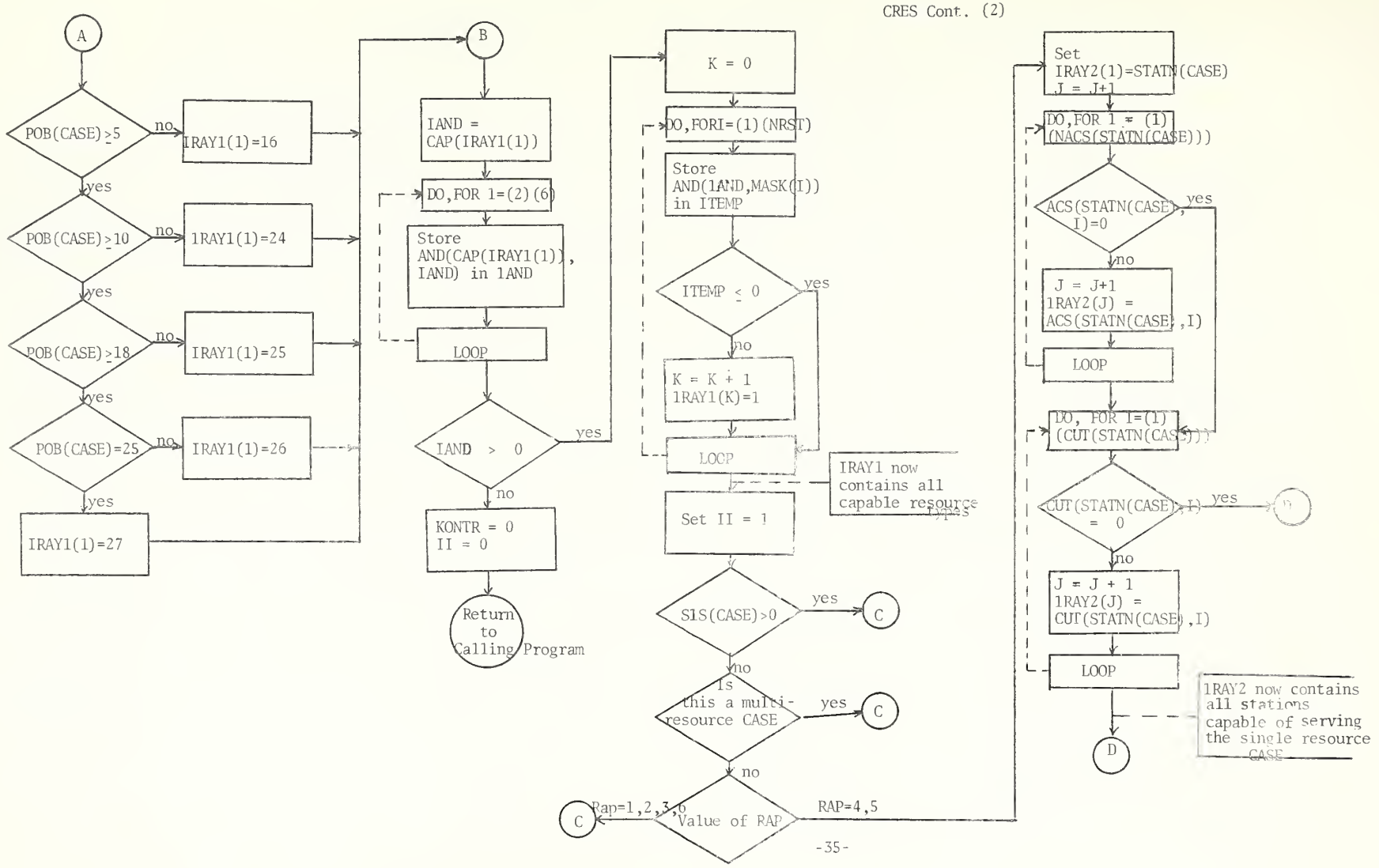

CALL CRES
II = 0 ?
no
KNT = 0 ?
no
CALL VEC
yes
Value of II is returned from CRES; II=0, no capable resource types; II=1, capable resource types do exist
yes
Value of KNT is returned from CRES; KNT=0, no capable resources; KNT≠ 0, KNT capable resources are in ROUE
Set ITOL = 3
Set ITOL = 4
Increase FAIL1 by 1
Increase FAIL2 by 1
File CASE in EXCS
File CASE in EXCS
Return to calling program
Return to calling program
CALL OSET
CALL RESAP
Value of IRES is returned from RESAP; IRES = 0, no available resource to serve CASE IRES ≠ 0, resource that will serve the CASE
IRES = 0 ?
CALL SERVE
Return to calling program
yes
NEED = 17 ?
Set ITOL=5
File CASE in EXCS
Is NEED an air escort?
no
Set TINO,REA, SIGNL; Increase NQUE by 1
File CASE in CQUE
Return to calling program
Return to calling program

-33-

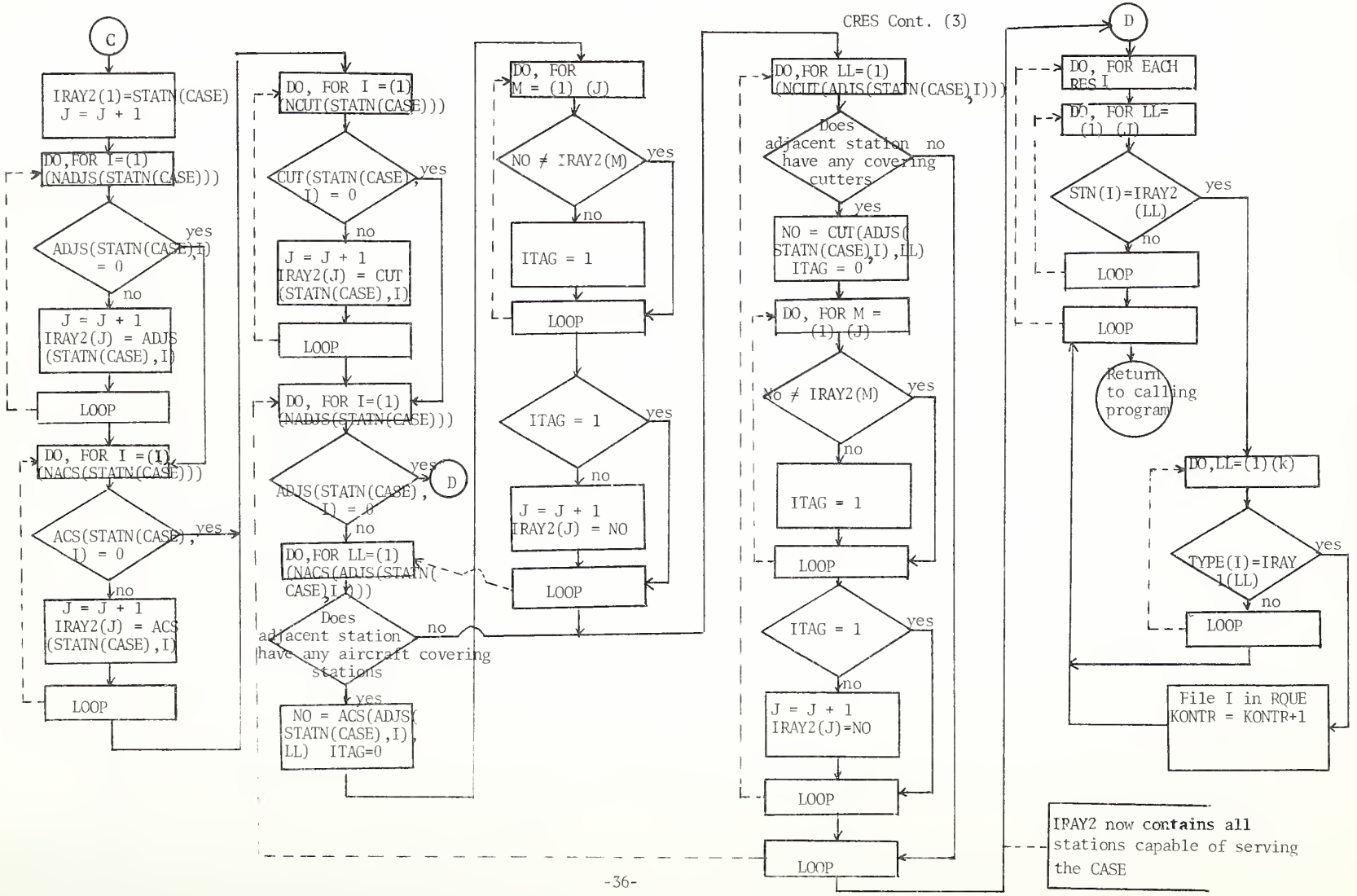
SUBROUTINE CRES



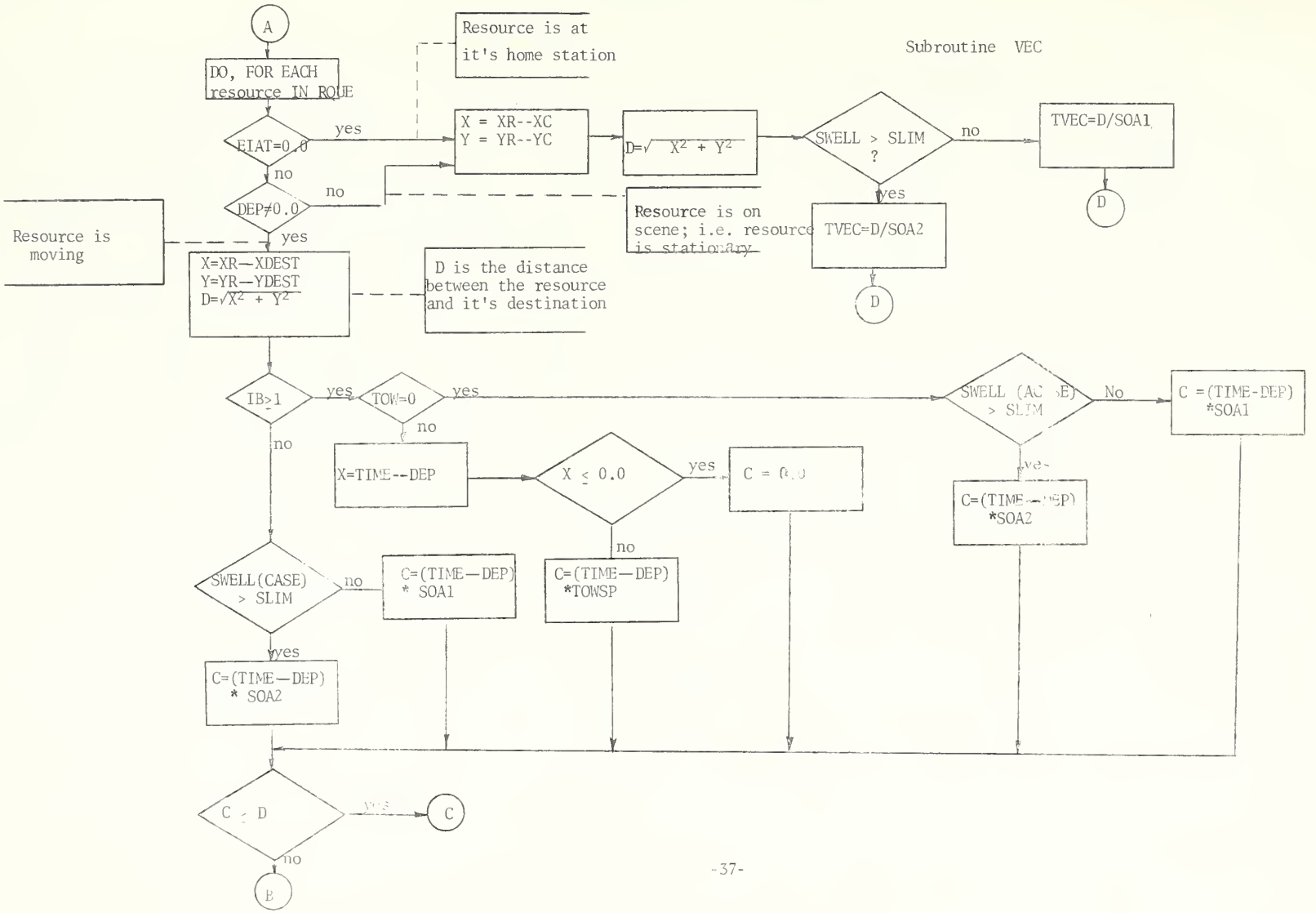
-34-


A
POB(CASE)≥5
no
IRAY1(1)=16
yes
POB(CASE)≥10
no
1RAY1(1)=24
yes
POB(CASE)≥18
no
IRAY1(1)=25
yes
POB(CASE)=25
no
IRAY1(1)=26
yes
IRAY1(1)=27
B
IAND = CAP(IRAY1(1))
DO,FOR 1=(2)(6)
Store AND(CAP(IRAY1(1)), IAND) in 1AND
LOOP
IAND > 0
yes
no
KONTR = 0
II = 0
Return to Calling Program
K = 0
DO,FORI=(1)(NRST)
Store AND(1AND,MASK(I)) in ITEMP
ITEMP ≤ 0
yes
no
K = K + 1
1RAY1(K)=1
LOOP
IRAY1 now contains all capable resources
Set II = 1
S1S(CASE)>0
yes
C
no
Is this a multi-resource CASE
yes
C
no
Value of RAP
RAP=1,2,3,6
C
RAP=4,5
Set IRAY2(1)=STATN(CASE) J = J+1
DO,FOR I = (1) (NACS(STATN(CASE)))
ACS(STATN(CASE), I)=0
yes
no
J = J+1 1RAY2(J) = ACS(STATN(CASE),I)
LOOP
DO, FOR I=(1) (CUT(STATN(CASE)))
CUT(STATN(CASE),I) = 0
yes
no
J = J + 1 1RAY2(J) = CUT(STATN(CASE),I)
LOOP
1RAY2 now contains all stations capable of serving the single resource CASE
D

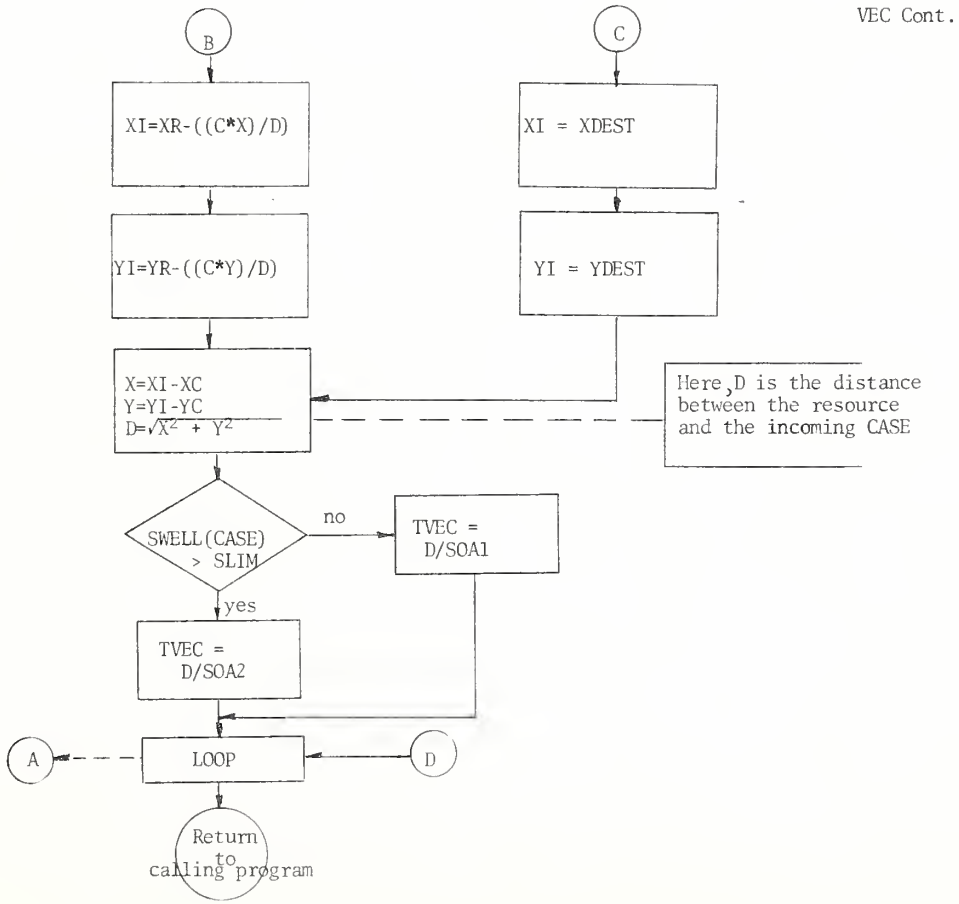
-35-


CRES Cont. (3)
C
IRAY2(1)=STATN(CASE)
J = J + 1
DO,FOR I=(1) (NADJS(STATN(CASE)))
ADJS(STATN(CASE),I) = 0
yes
no
J = J + 1
IRAY2(J) = ADJS (STATN(CASE),I)
LOOP
DO, FOR I =(1) (NACS(STATN(CASE)))
ACS(STATN(CASE), I) = 0
yes
no
J = J + 1
IRAY2(J) = ACS (STATN(CASE),I)
LOOP
DO, FOR I =(1) (NCUT(STATN(CASE)))
CUT(STATN(CASE), I) = 0
yes
no
J = J + 1
IRAY2(J) = CUT (STATN(CASE),I)
LOOP
DO, FOR I=(1) (NADJS(STATN(CASE)))
ADJS(STATN(CASE), I) = 0
yes
no
D
DO,FOR LL=(1) (NACS(ADJS(STATN(CASE),I)))
Does adjacent station have any aircraft covering stations
no
yes
NO = ACS(ADJS(STATN(CASE),I), LL) ITAG=0
DO, FOR M = (1) (J)
NO ≠ IRAY2(M)
yes
no
ITAG = 1
LOOP
ITAG = 1
yes
no
J = J + 1
IRAY2(J) = NO
LOOP
DO,FOR LL=(1) (NCUT(ADJS(STATN(CASE),I)))
Does adjacent station have any covering cutters
no
yes
NO = CUT(ADJS(STATN(CASE),I),LL) ITAG = 0
DO, FOR M = (1) (J)
NO ≠ IRAY2(M)
yes
no
ITAG = 1
LOOP
ITAG = 1
yes
no
J = J + 1
IRAY2(J)=NO
LOOP
LOOP
D
DO, FOR EACH RES I
DO, FOR LL= (1) (J)
STN(I)=IRAY2 (LL)
yes
no
LOOP
LOOP
Return to calling program
DO,LL=(1)(k)
TYPE(I)=IRAY 1(LL)
yes
no
LOOP
File I in RQUE
KONTR = KONTR+1
IRAY2 now contains all stations capable of serving the CASE

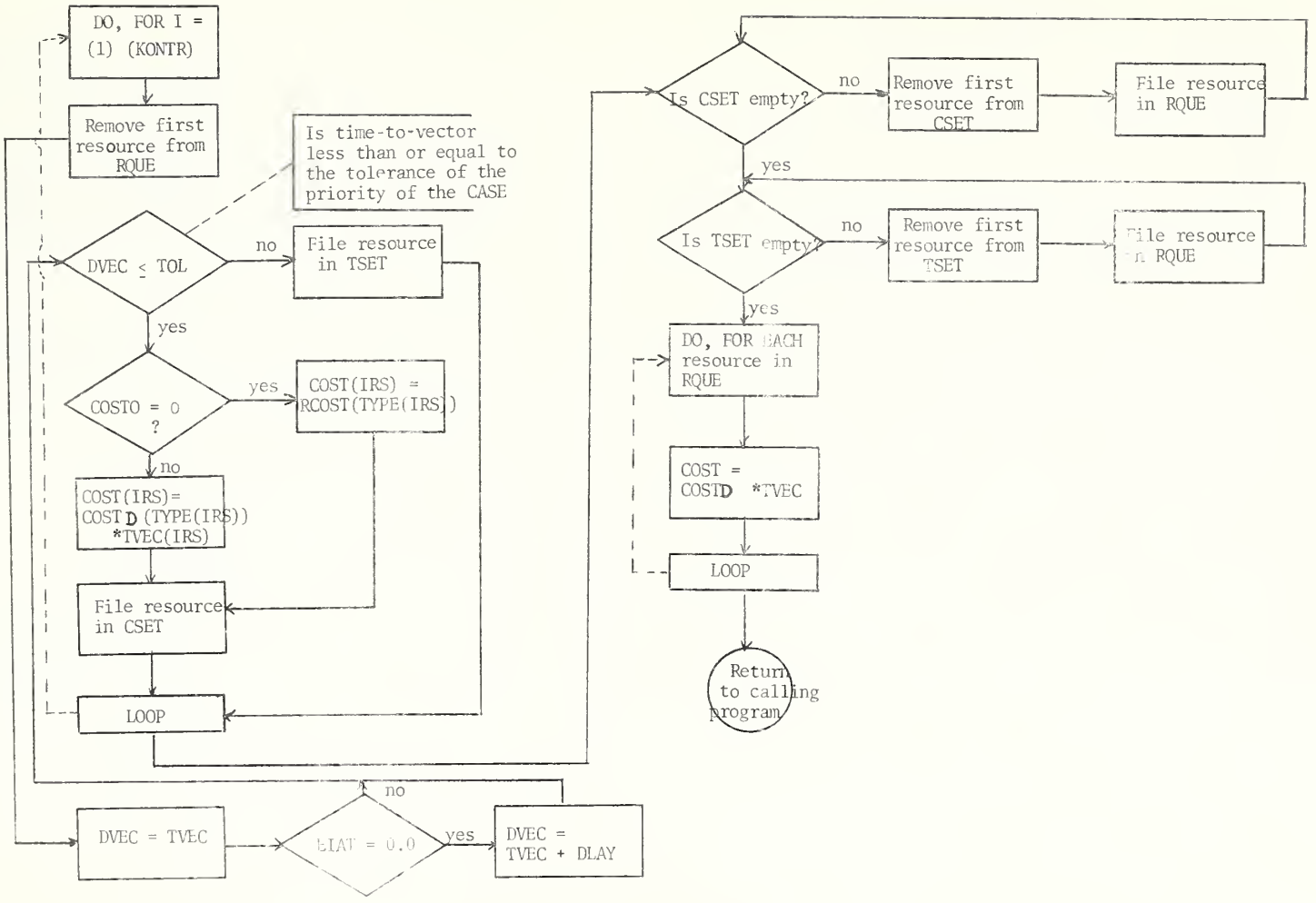
-36-


Subroutine VEC
A
DO, FOR EACH resource IN RQUE
EIAT=0.0
yes
no
DEP≠0.0
no
yes
Resource is at it's home station
X = XR--XC
Y = YR--YC
D=√X² + Y²
SWELL > SLIM ?
no
yes
TVEC=D/SOA1
D
Resource is on scene; i.e. resource is stationary
TVEC=D/SOA2
D
Resource is moving
X=XR—XDEST
Y=YR—YDEST
D=√X² + Y²
D is the distance between the resource and it's destination
IB≥1
yes
no
TOW=0
yes
no
X=TIME--DEP
X ≤ 0.0
yes
C = 0.0
no
C=(TIME—DEP) *TOWSP
SWELL (AC >E) > SLIM
No
C =(TIME-DEP) *SOA1
yes
C=(TIME—DEP) *SOA2
SWELL(CASE) > SLIM
no
C=(TIME—DEP) * SOA1
yes
C=(TIME—DEP) * SOA2
C ≤ D
yes
C
no
B

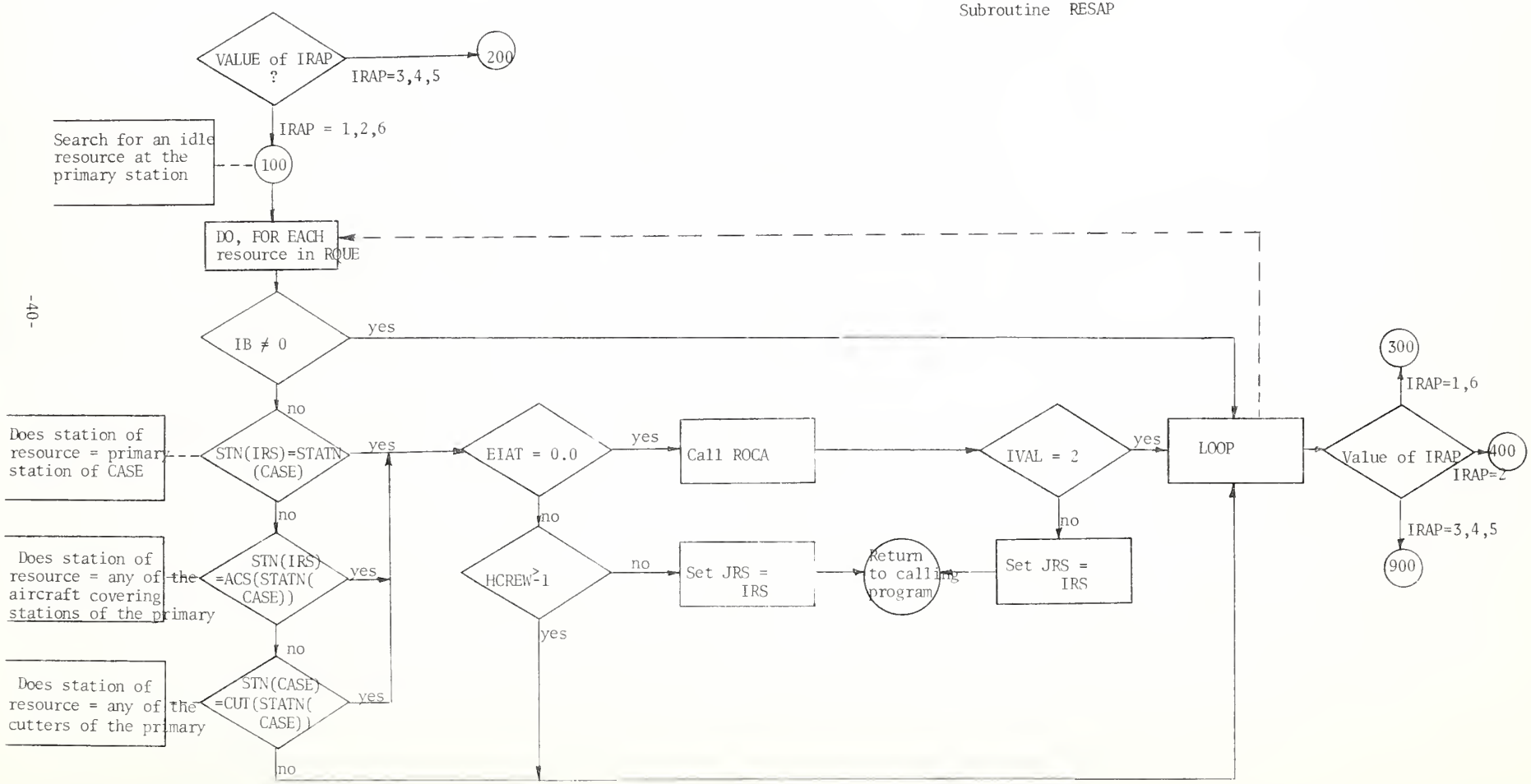
-37-

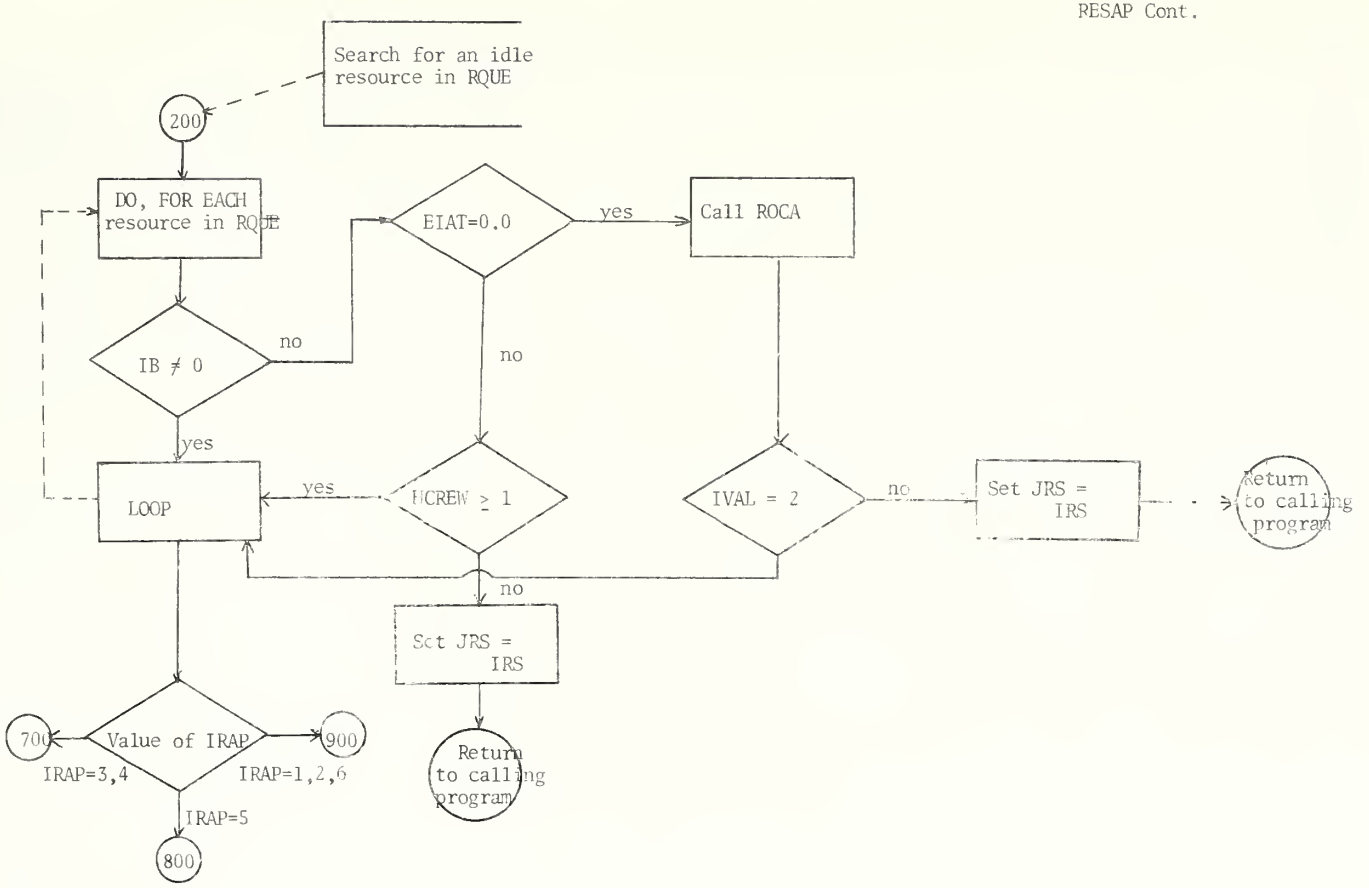

B
C
XI=XR-((C*X)/D)
XI = XDEST
YI=YR-((C*Y)/D)
YI = YDEST
X=XI-XC
Y=YI-YC
D=√X² + Y²
Here, D is the distance between the resource and the incoming CASE
SWELL(CASE) > SLIM
no
TVEC = D/SOA1
yes
TVEC = D/SOA2
A
LOOP
D
Return to calling program

-38-

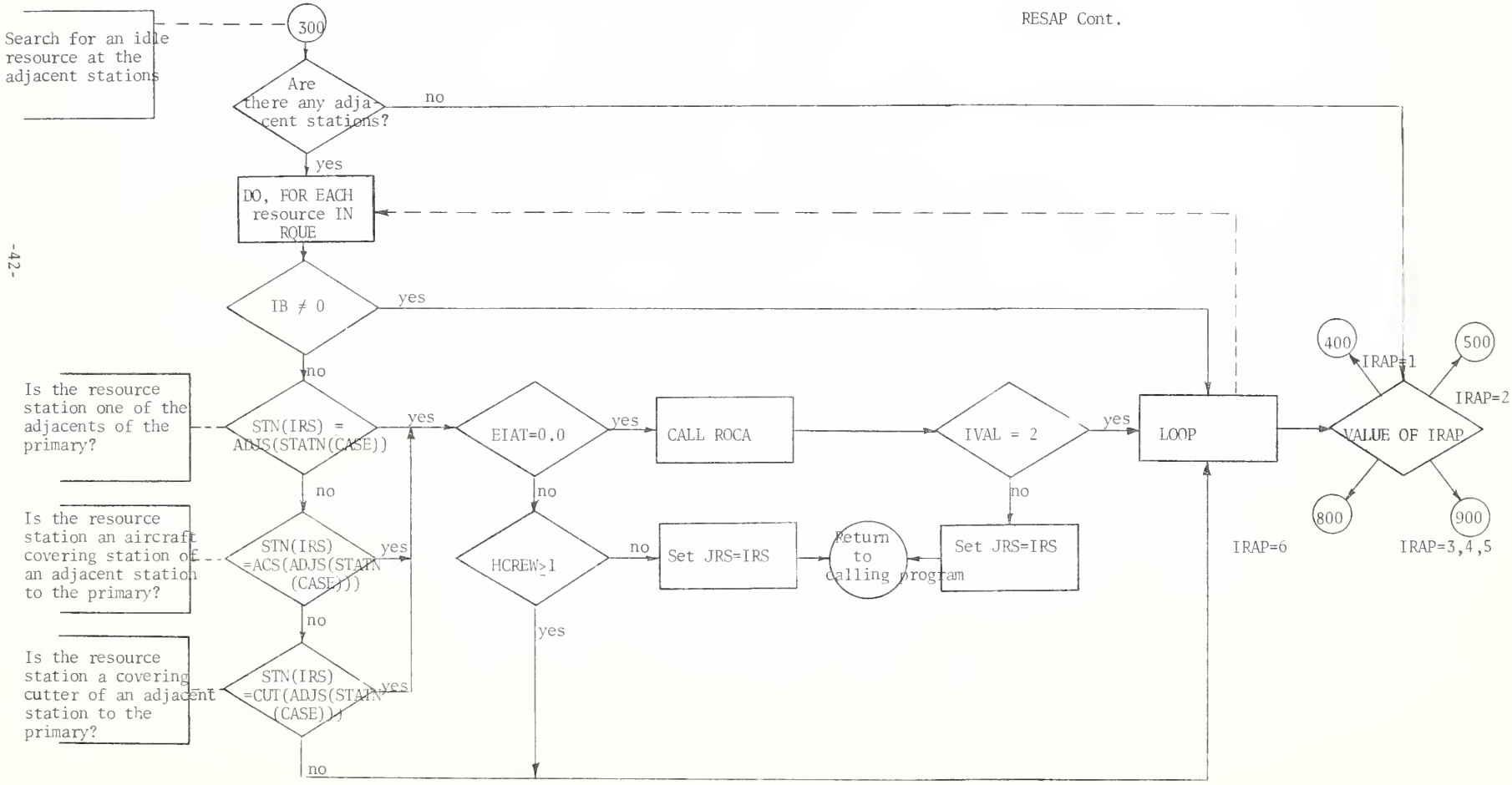

Subroutine OSET
DO, FOR I = (1) (KONTR)
Remove first resource from RQUE
Is time-to-vector less than or equal to the tolerance of the priority of the CASE
DVEC ≤ TOL
no
File resource in TSET
yes
COSTO = 0 ?
yes
COST(IRS) = RCOST(TYPE(IRS))
no
COST(IRS)= COSTD (TYPE(IRS)) *TVEC(IRS)
File resource in CSET
LOOP
DVEC = TVEC
BIAT = 0.0
no
yes
DVEC = TVEC + DLAY
Is CSET empty?
no
Remove first resource from CSET
File resource in RQUE
yes
Is TSET empty?
no
Remove first resource from TSET
File resource in RQUE
yes
DO, FOR EACH resource in RQUE
COST = COSTD *TVEC
LOOP
Return to calling program

Subroutine RESAP

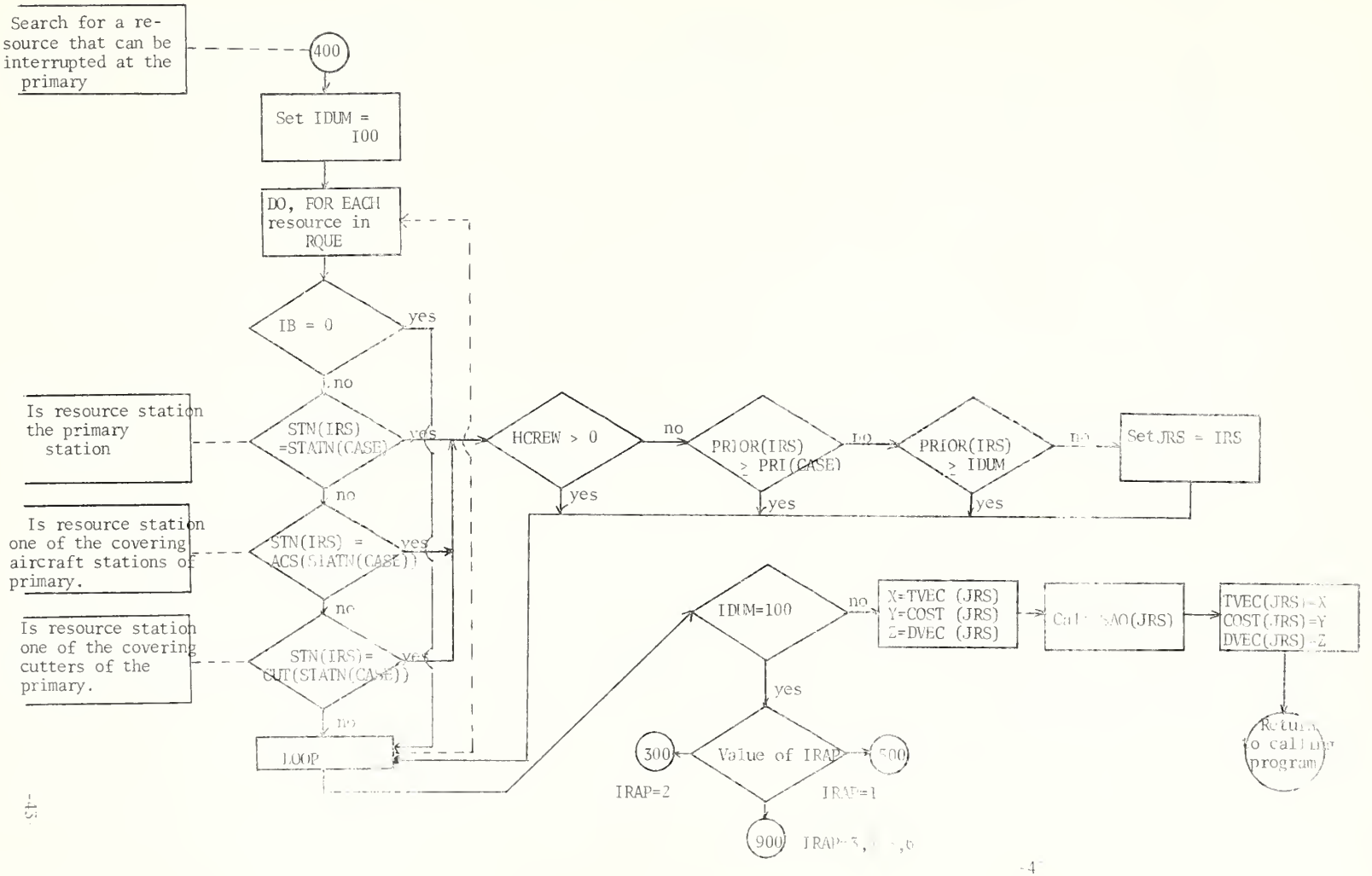


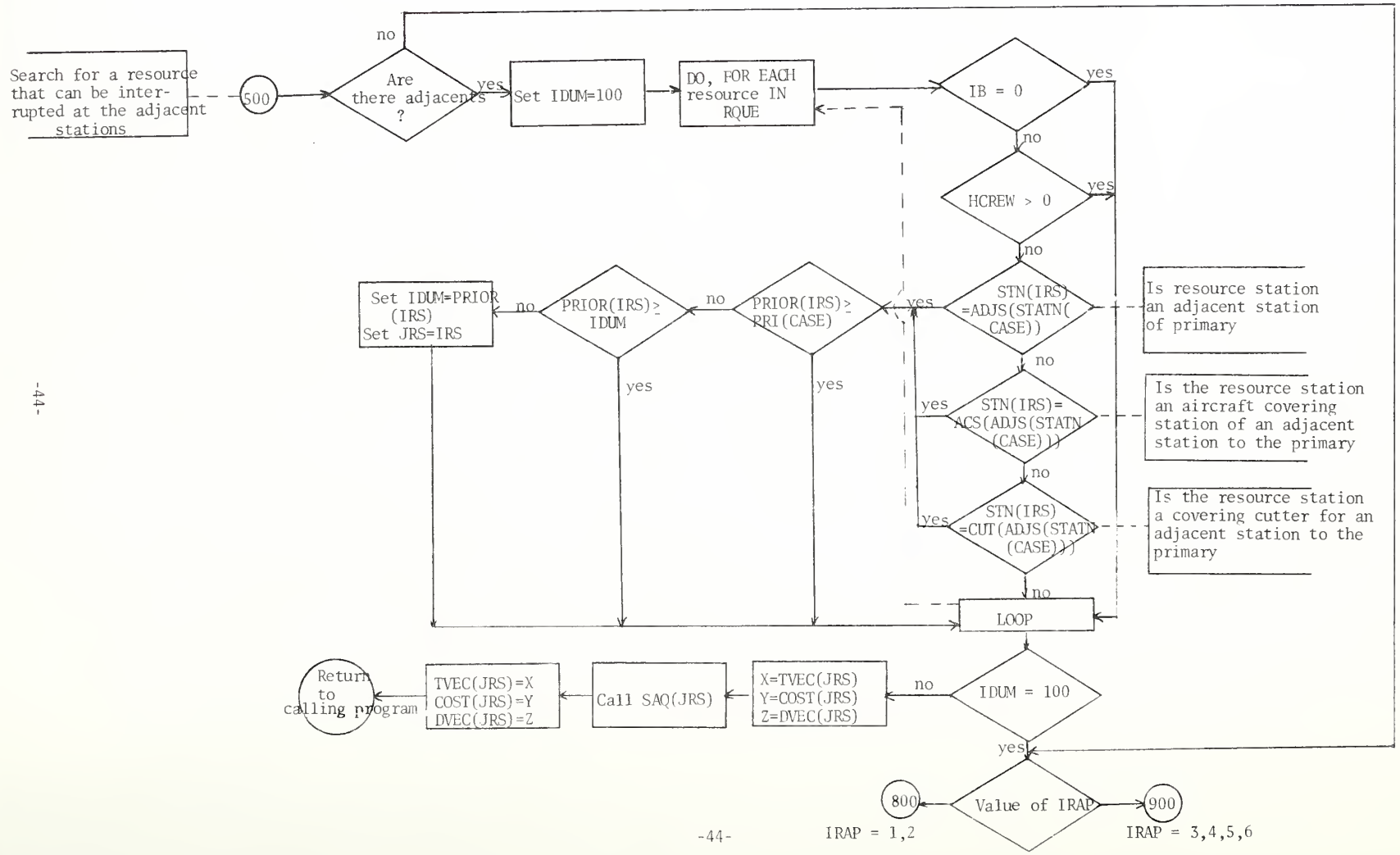

Search for an idle resource in RQUE
200
DO, FOR EACH resource in RQUE
EIAT=0.0
yes
Call ROCA
IB ≠ 0
no
no
yes
LOOP
yes
NCREW ≥ 1
IVAL = 2
no
Set JRS = IRS
Return to calling program
no
Set JRS = IRS
Return to calling program
700
Value of IRAP
900
IRAP=3,4
IRAP=1,2,6
IRAP=5
800

-41-

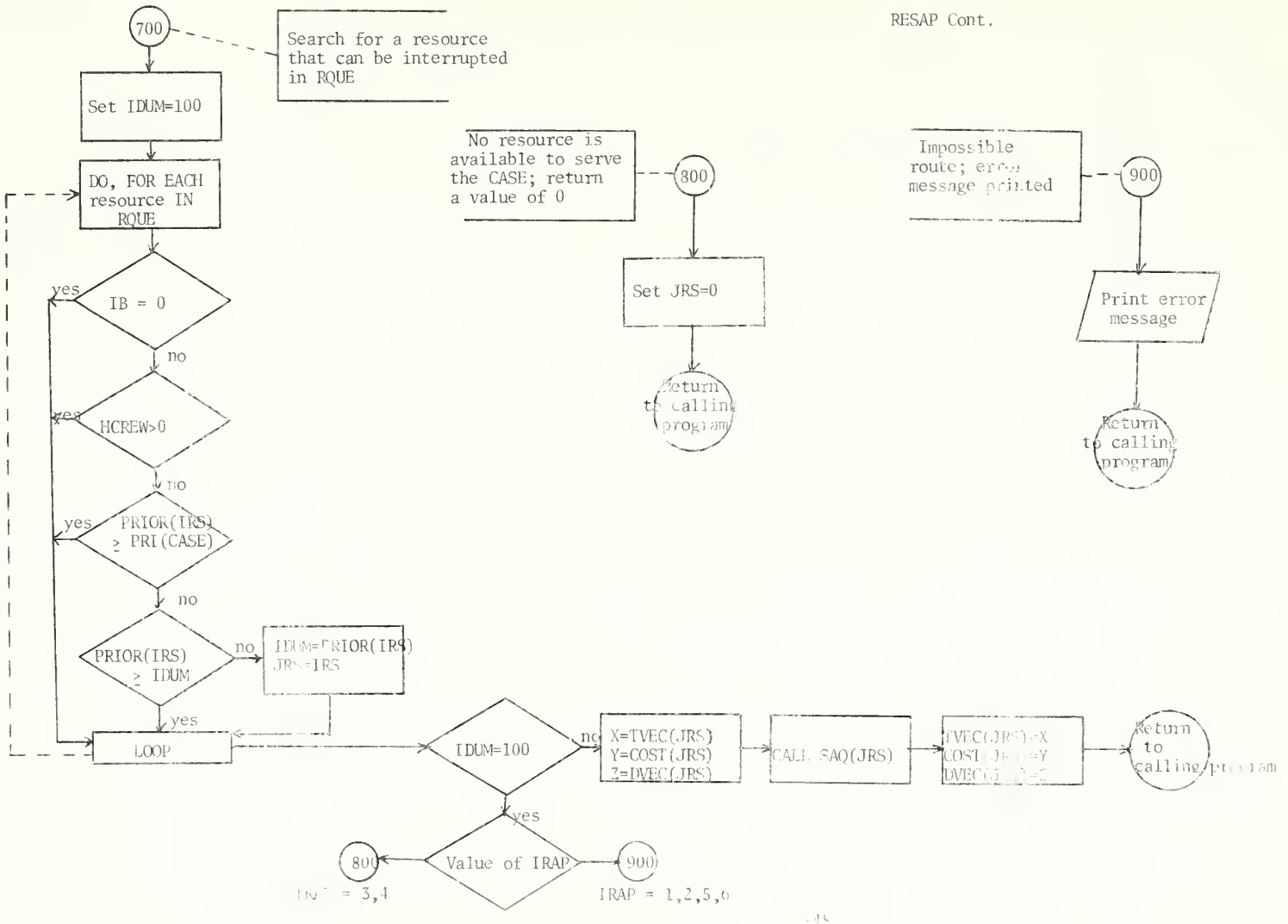

Search for an idle resource at the adjacent stations
300
Are there any adjacent stations?
no
yes
DO, FOR EACH resource IN RQUE
IB ≠ 0
yes
no
Is the resource station one of the adjacents of the primary?
STN(IRS) = ADJS(STATN(CASE))
yes
EIAT=0.0
yes
CALL ROCA
IVAL = 2
yes
LOOP
no
no
no
Is the resource station an aircraft covering station of an adjacent station to the primary?
STN(IRS) =ACS(ADJS(STATN (CASE)))
yes
HCREW≥1
no
Set JRS=IRS
Return to calling program
Set JRS=IRS
yes
no
Is the resource station a covering cutter of an adjacent station to the primary?
STN(IRS) =CUT(ADJS(STATN (CASE)))
yes
no
400
IRAP=1
500
IRAP=2
VALUE OF IRAP
800
900
IRAP=6
IRAP=3,4,5

-42-

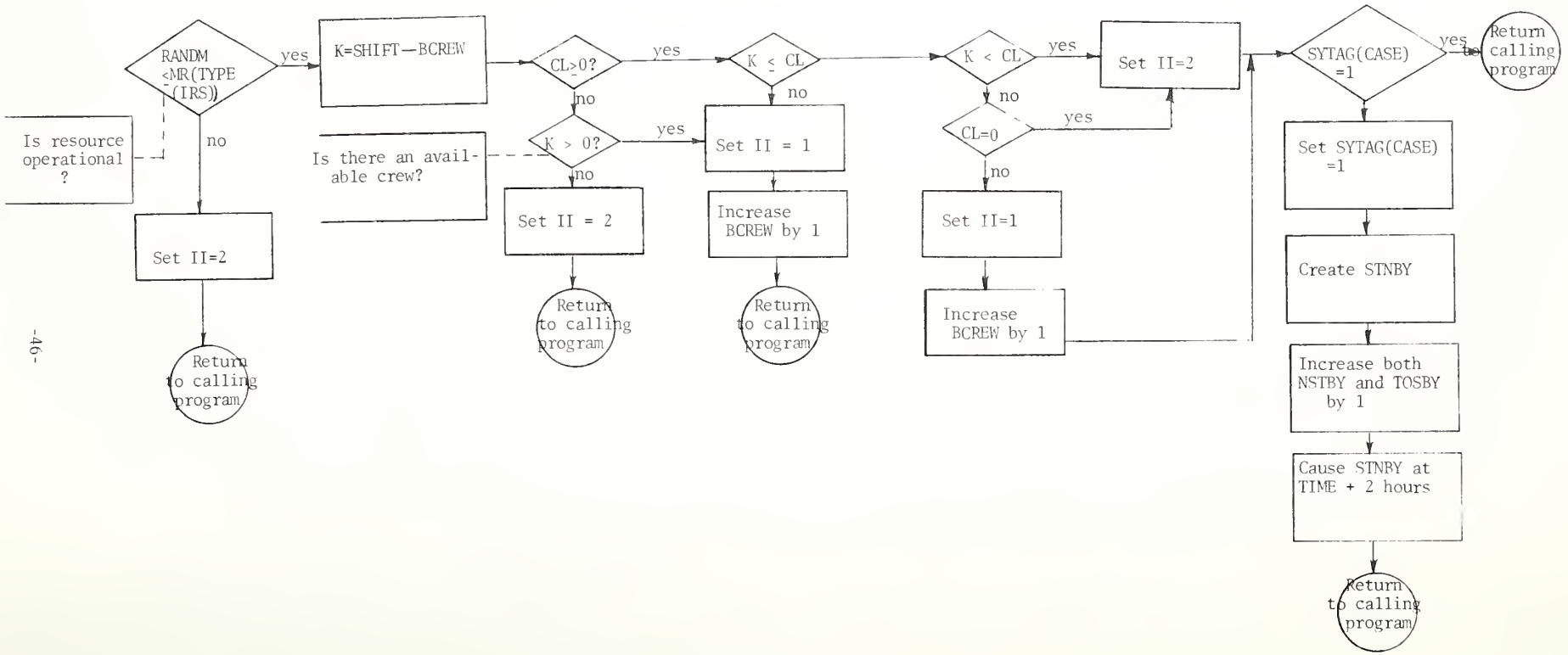

Search for a resource that can be interrupted at the primary
400
Set IDUM = I00
DO, FOR EACH resource in RQUE
IB = 0
yes
no
Is resource station the primary station
STN(IRS) =STATN(CASE)
yes
no
Is resource station one of the covering aircraft stations of primary.
STN(IRS) = ACS(STATN(CASE))
yes
no
Is resource station one of the covering cutters of the primary.
STN(IRS)= CUT(STATN(CASE))
yes
no
LOOP
HCREW > 0
no
yes
PRIOR(IRS) ≥ PRI(CASE)
no
yes
PRIOR(IRS) > IDUM
no
yes
Set JRS = IRS
IDUM=100
no
yes
X=TVEC (JRS)
Y=COST (JRS)
Z=DVEC (JRS)
Call SAO(JRS)
TVEC(JRS)=X
COST(JRS)=Y
DVEC(JRS)=Z
Return to calling program
Value of IRAP
300
IRAP=2
500
IRAP=1
900
IRAP=3, , ,6


Search for a resource that can be interrupted at the adjacent stations
500
Are there adjacents ?
no
yes
Set IDUM=100
DO, FOR EACH resource IN RQUE
IB = 0
yes
no
HCREW > 0
yes
no
STN(IRS) =ADJS(STATN(CASE))
Is resource station an adjacent station of primary
yes
no
STN(IRS)= ACS(ADJS(STATN (CASE)))
Is the resource station an aircraft covering station of an adjacent station to the primary
yes
no
STN(IRS) =CUT(ADJS(STATN (CASE)))
Is the resource station a covering cutter for an adjacent station to the primary
yes
no
PRIOR(IRS)≥ PRI(CASE)
yes
no
PRIOR(IRS)≥ IDUM
yes
no
Set IDUM=PRIOR (IRS) Set JRS=IRS
LOOP
IDUM = 100
no
yes
X=TVEC(JRS) Y=COST(JRS) Z=DVEC(JRS)
Call SAQ(JRS)
TVEC(JRS)=X COST(JRS)=Y DVEC(JRS)=Z
Return to calling program
Value of IRAP
800
IRAP = 1,2
900
IRAP = 3,4,5,6

-44-

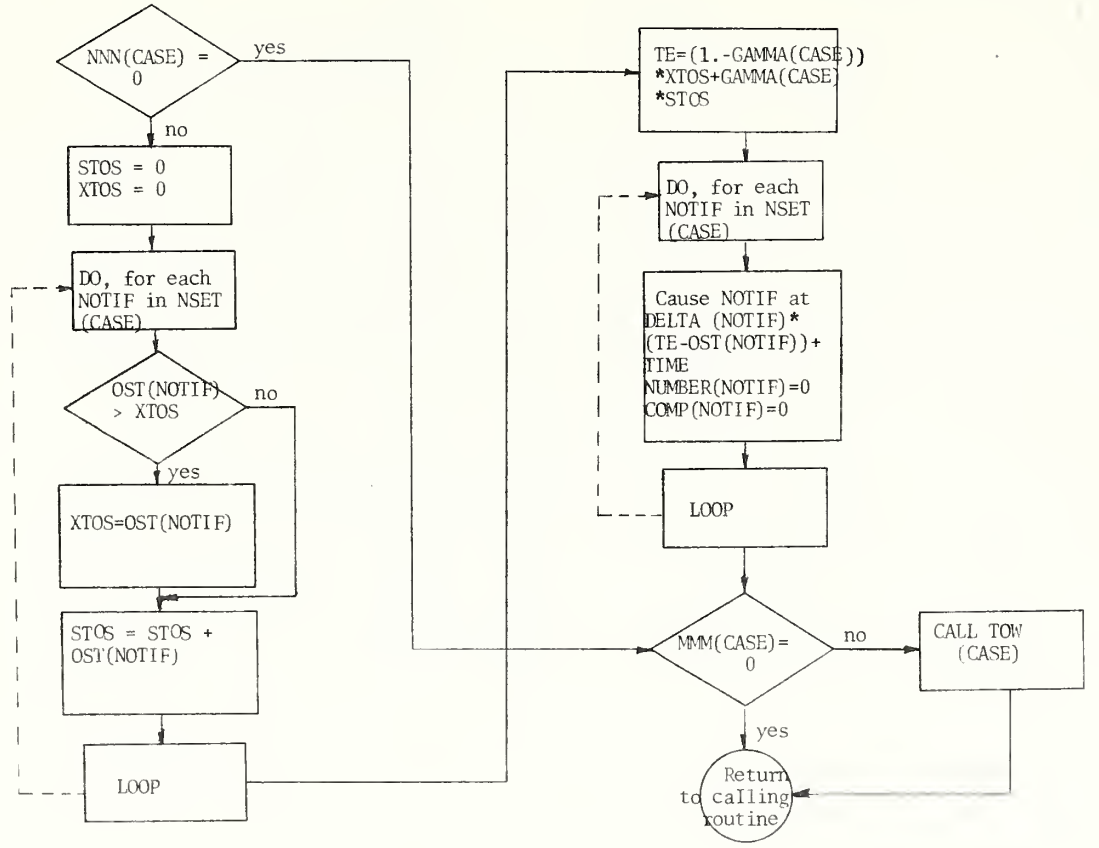

700
Search for a resource that can be interrupted in RQUE
Set IDUM=100
DO, FOR EACH resource IN RQUE
IB = 0
yes
no
HCREW>0
yes
no
PRIOR(IRS) ≥ PRI(CASE)
yes
no
PRIOR(IRS) ≥ IDUM
no
IDUM=PRIOR(IRS)
JRS=IRS
yes
LOOP
IDUM=100
no
X=TVEC(JRS)
Y=COST(JRS)
Z=DVEC(JRS)
CALL SAQ(JRS)
TVEC(JRS)=X
COST(JRS)=Y
DVEC(JRS)=Z
Return to calling program
yes
Value of IRAP
800
IRAP = 3,4
900
IRAP = 1,2,5,6
No resource is available to serve the CASE; return a value of 0
800
Set JRS=0
Return to calling program
Impossible route; error message printed
900
Print error message
Return to calling program

Subroutine ROCA

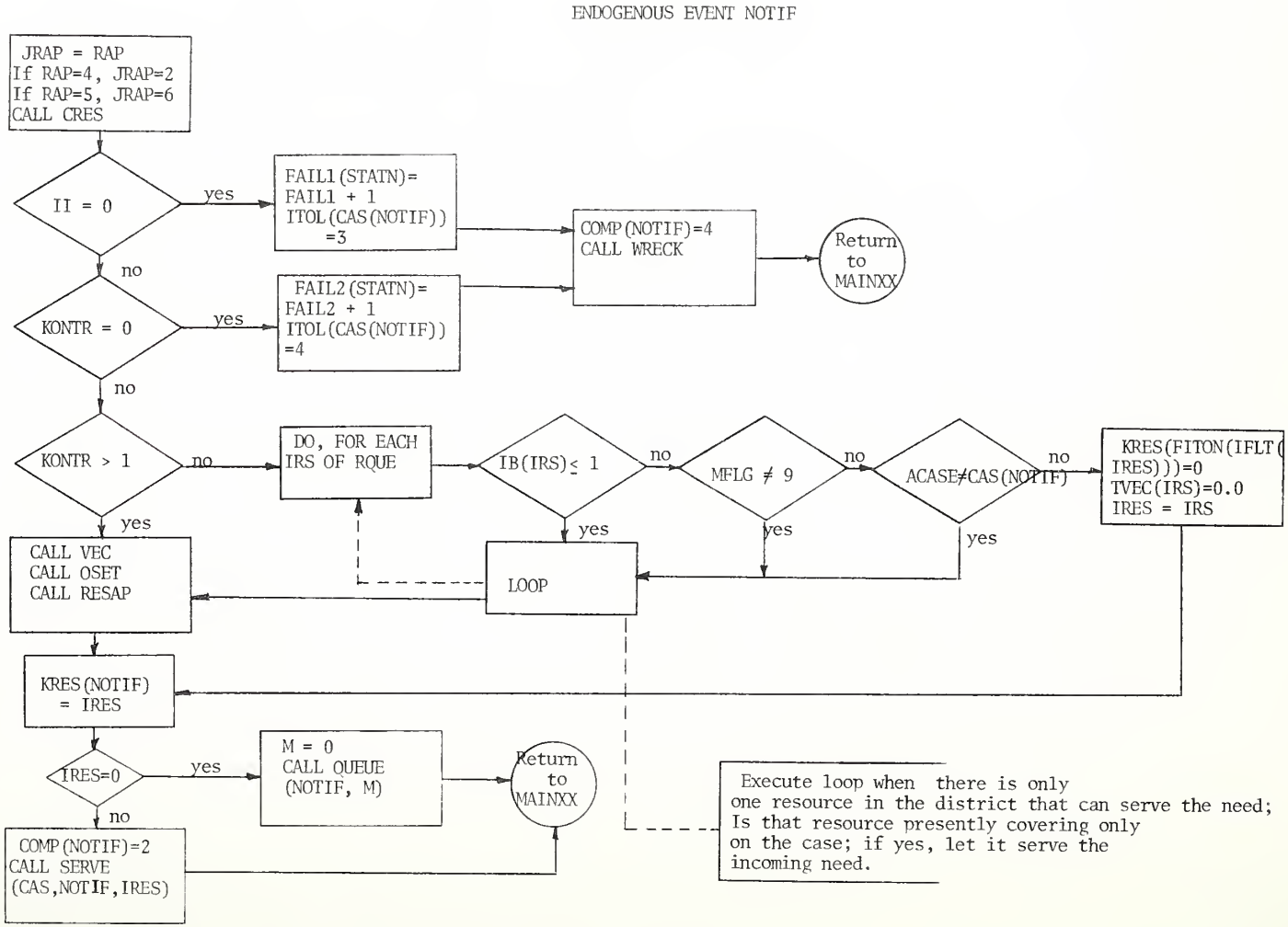


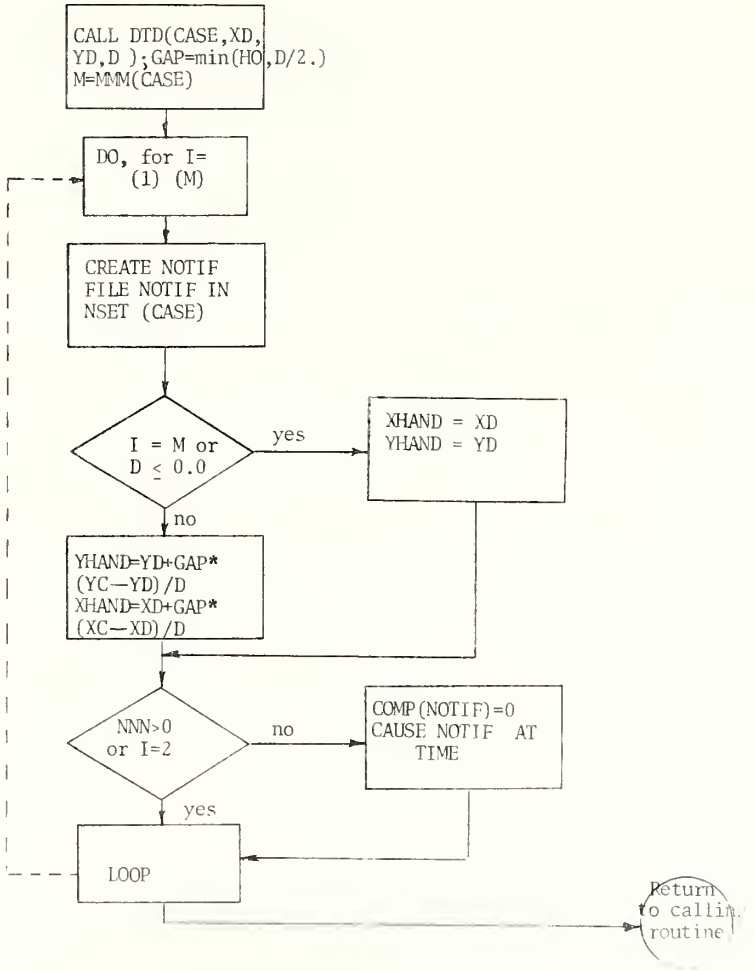
-46-

SUBROUTINE MRAS (CASE)

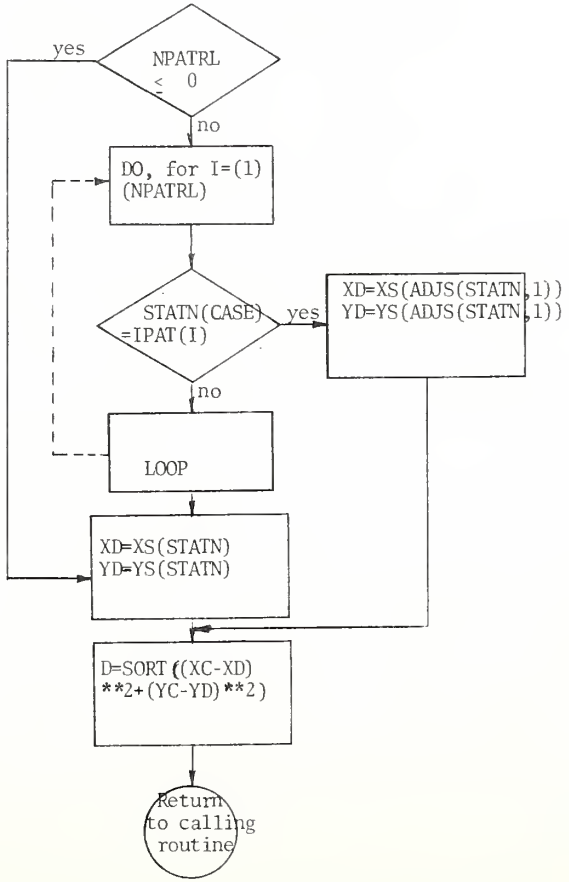


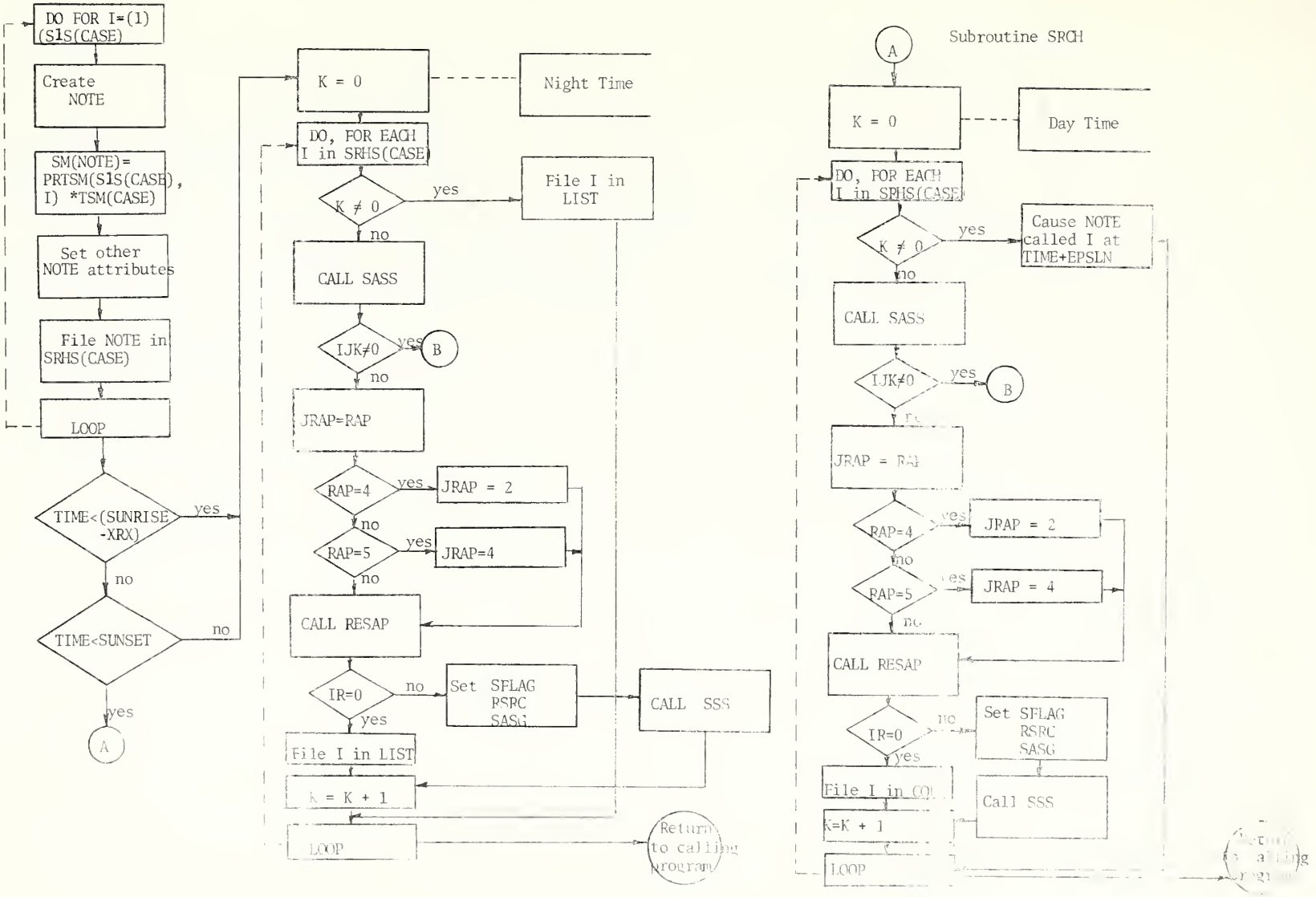
-47-

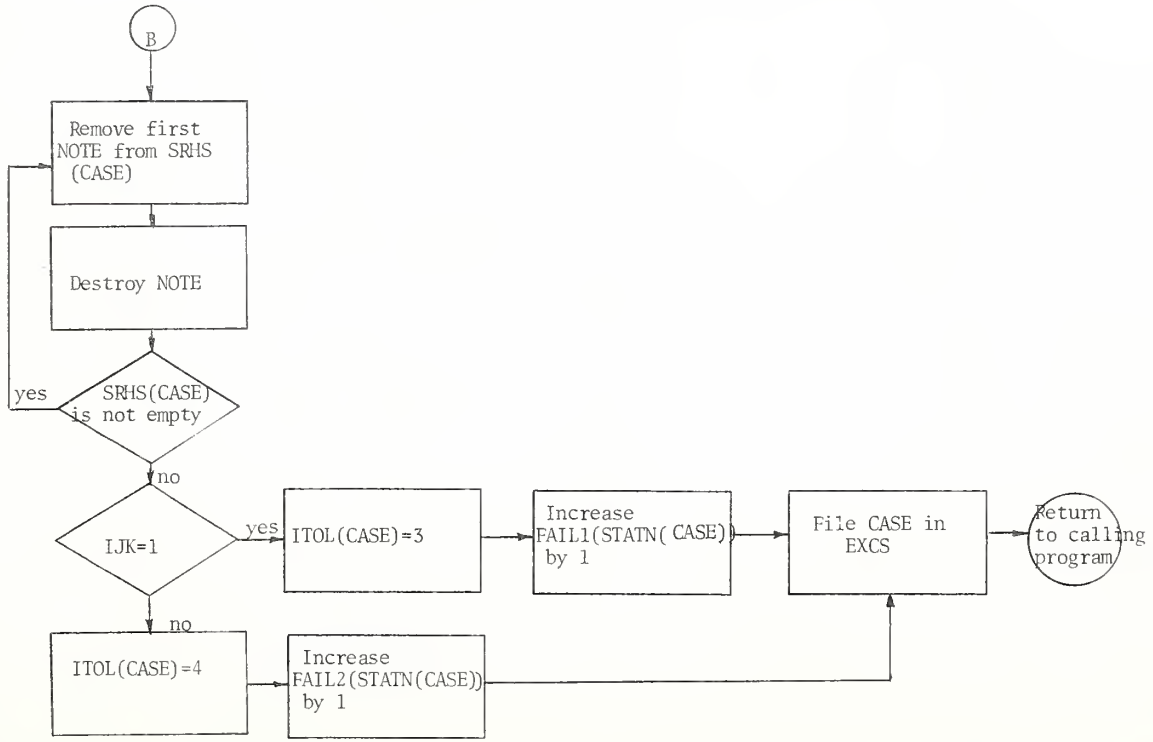

ENDOGENOUS EVENT NOTIF
JRAP = RAP
If RAP=4, JRAP=2
If RAP=5, JRAP=6
CALL CRES
II = 0
yes
FAIL1(STATN)=
FAIL1 + 1
ITOL(CAS(NOTIF))
=3
COMP(NOTIF)=4
CALL WRECK
Return
to
MAINXX
no
KONTR = 0
yes
FAIL2(STATN)=
FAIL2 + 1
ITOL(CAS(NOTIF))
=4
no
KONTR > 1
no
DO, FOR EACH
IRS OF RQUE
IB(IRS)≤ 1
no
MFLG ≠ 9
no
ACASE≠CAS(NOTIF)
no
KRES(FITON(IFLT(
IRES)))=0
TVEC(IRS)=0.0
IRES = IRS
yes
yes
yes
yes
CALL VEC
CALL OSET
CALL RESAP
LOOP
KRES(NOTIF)
= IRES
IRES=0
yes
M = 0
CALL QUEUE
(NOTIF, M)
Return
to
MAINXX
no
COMP(NOTIF)=2
CALL SERVE
(CAS,NOTIF,IRES)
Execute loop when there is only
one resource in the district that can serve the need;
Is that resource presently covering only
on the case; if yes, let it serve the
incoming need.

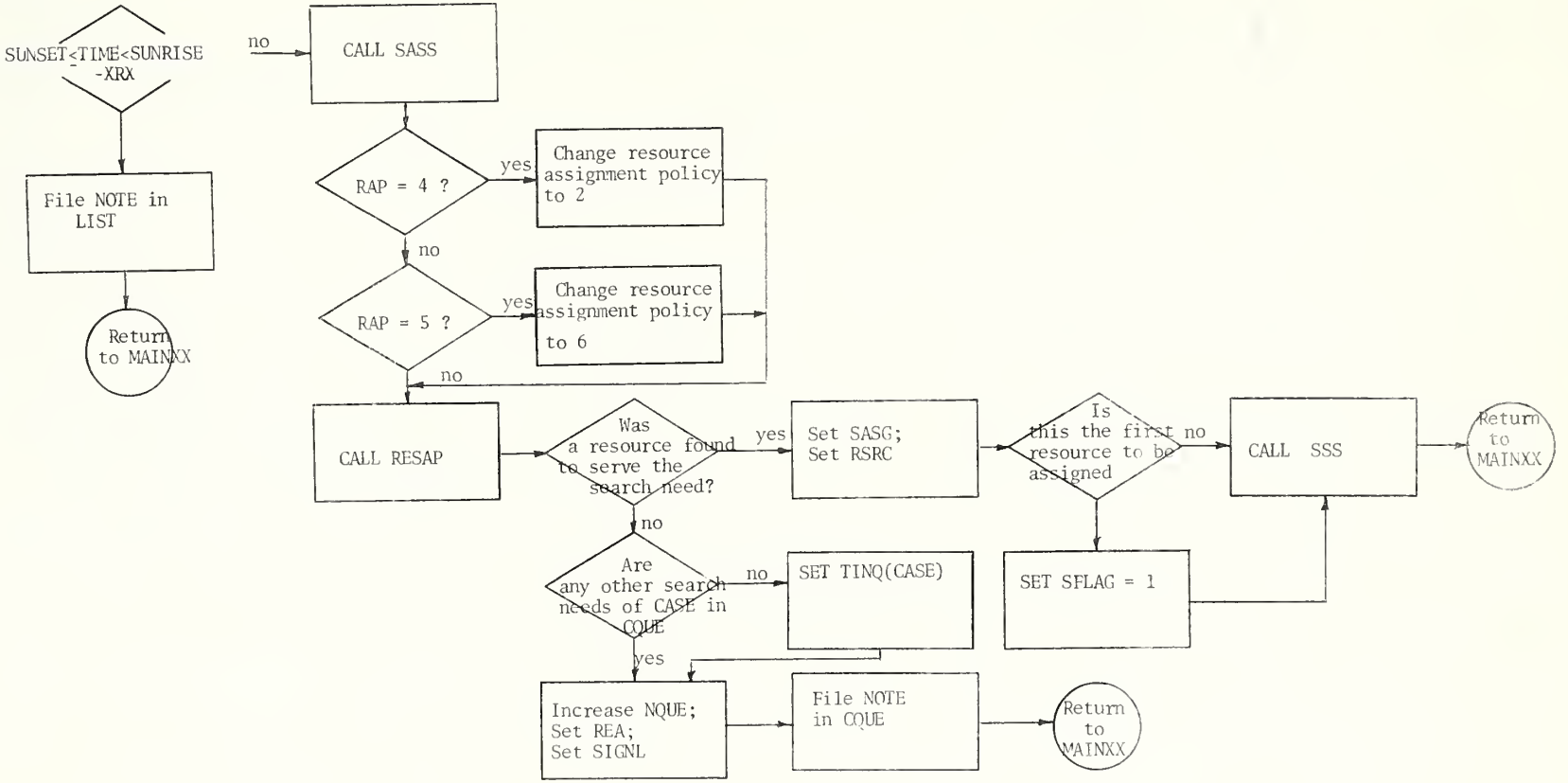

CALL DTD(CASE,XD,
YD,D);GAP=min(HO,D/2.)
M=MMM(CASE)
DO, for I=
(1) (M)
CREATE NOTIF
FILE NOTIF IN
NSET (CASE)
I = M or
D ≤ 0.0
yes
XHAND = XD
YHAND = YD
no
YHAND=YD+GAP*
(YC—YD)/D
XHAND=XD+GAP*
(XC—XD)/D
NNN>0
or I=2
no
COMP(NOTIF)=0
CAUSE NOTIF AT
TIME
yes
LOOP
Return
to callin
routine

SUBROUTINE DTD(CASE,XD,YD,D)

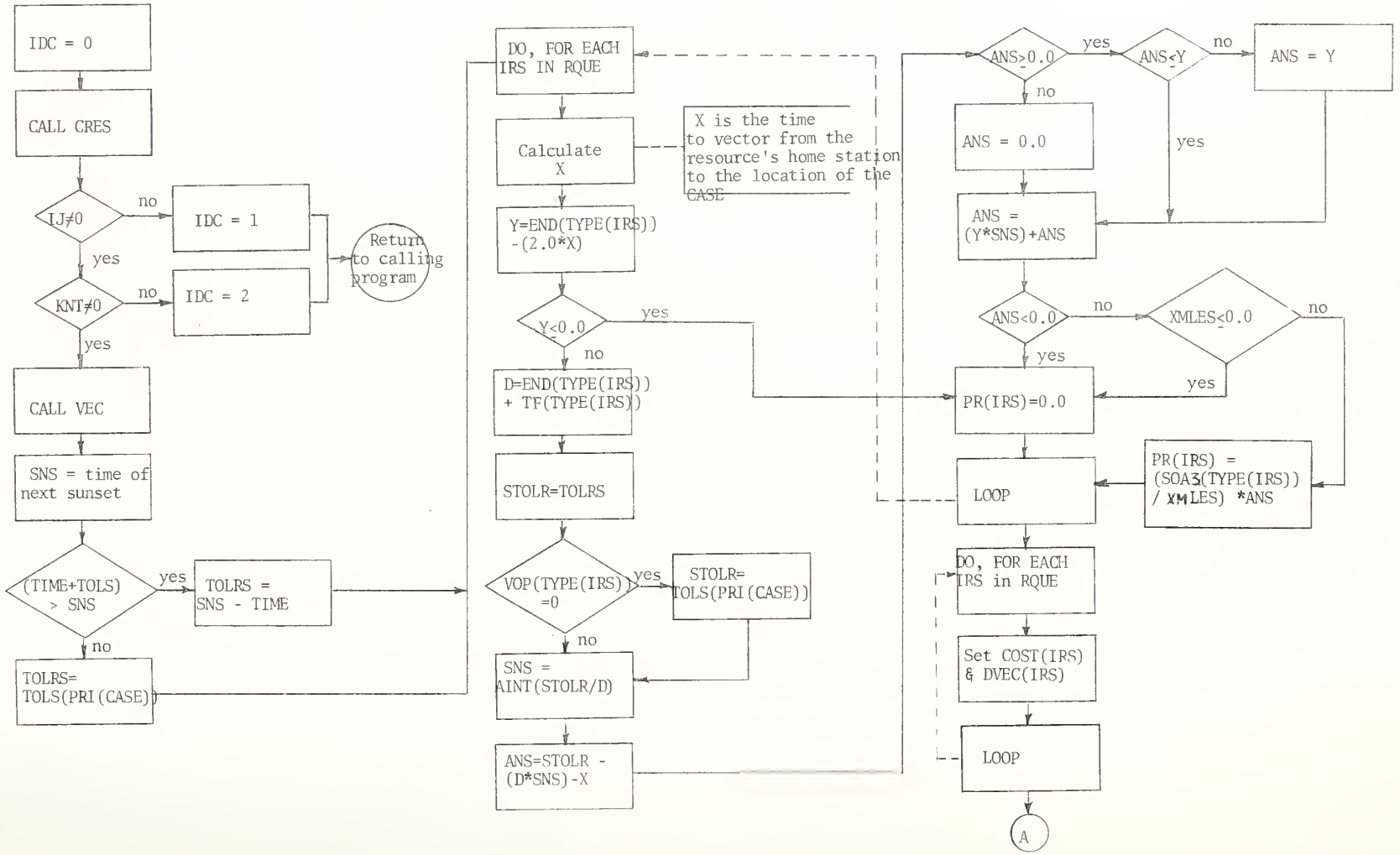


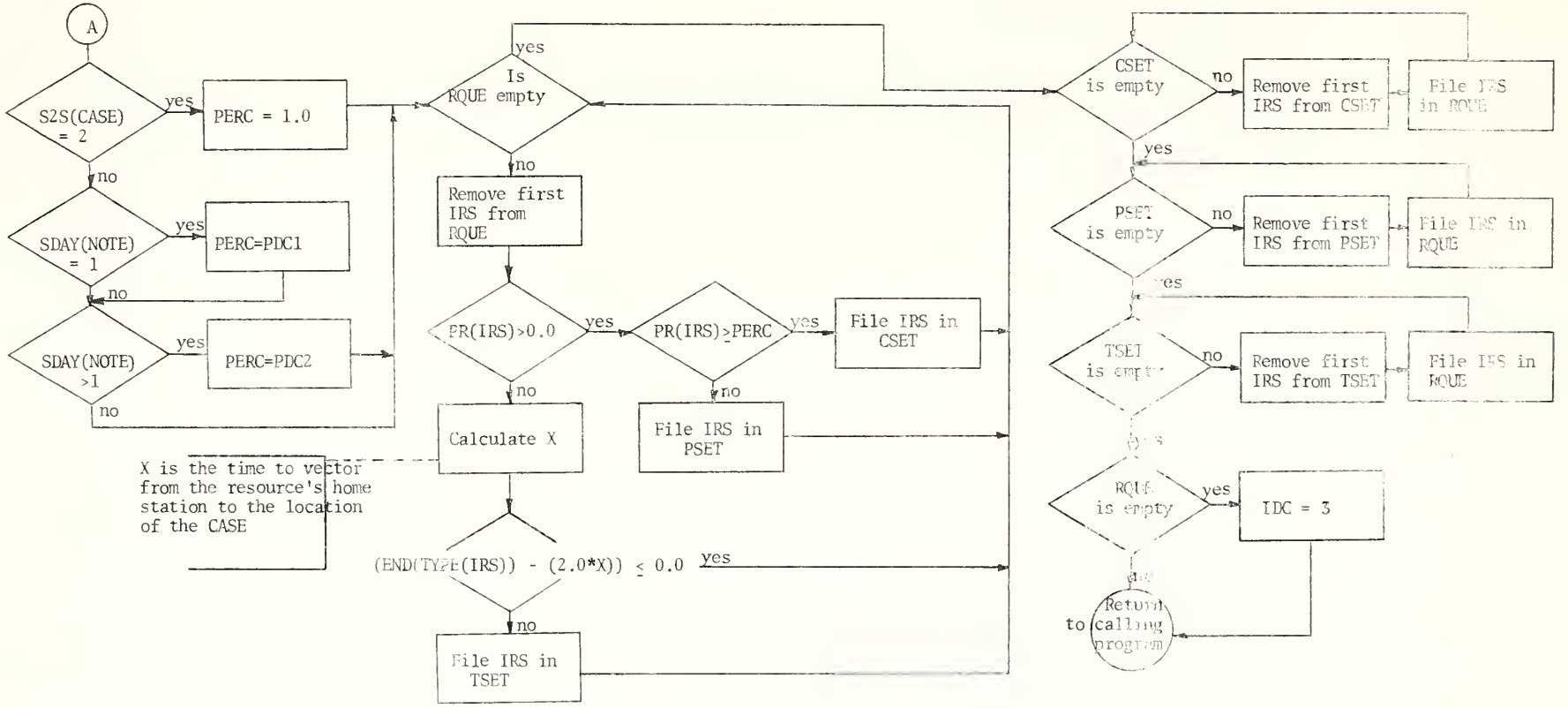

DO FOR I=(1) (S1S(CASE)
Create NOTE
SM(NOTE)= PRTSM(S1S(CASE), I) *TSM(CASE)
Set other NOTE attributes
File NOTE in SRHS(CASE)
LOOP
TIME<(SUNRISE -XRX)
yes
no
TIME<SUNSET
no
yes
A
K = 0
Night Time
DO, FOR EACH I in SRHS(CASE)
K ≠ 0
yes
File I in LIST
no
CALL SASS
IJK≠0
yes
B
no
JRAP=RAP
RAP=4
yes
JRAP = 2
no
RAP=5
yes
JRAP=4
no
CALL RESAP
IR=0
no
Set SFLAG RSRC SASG
yes
File I in LIST
K = K + 1
LOOP
CALL SSS
Return to calling program
Subroutine SRCH
A
K = 0
Day Time
DO, FOR EACH I in SRHS(CASE)
K ≠ 0
yes
Cause NOTE called I at TIME+EPSLN
no
CALL SASS
IJK≠0
yes
B
JRAP = RAP
RAP=4
yes
JRAP = 2
no
RAP=5
yes
JRAP = 4
no
CALL RESAP
IR=0
no
Set SFLAG RSRC SASG
yes
File I in COM
K=K + 1
LOOP
Call SSS

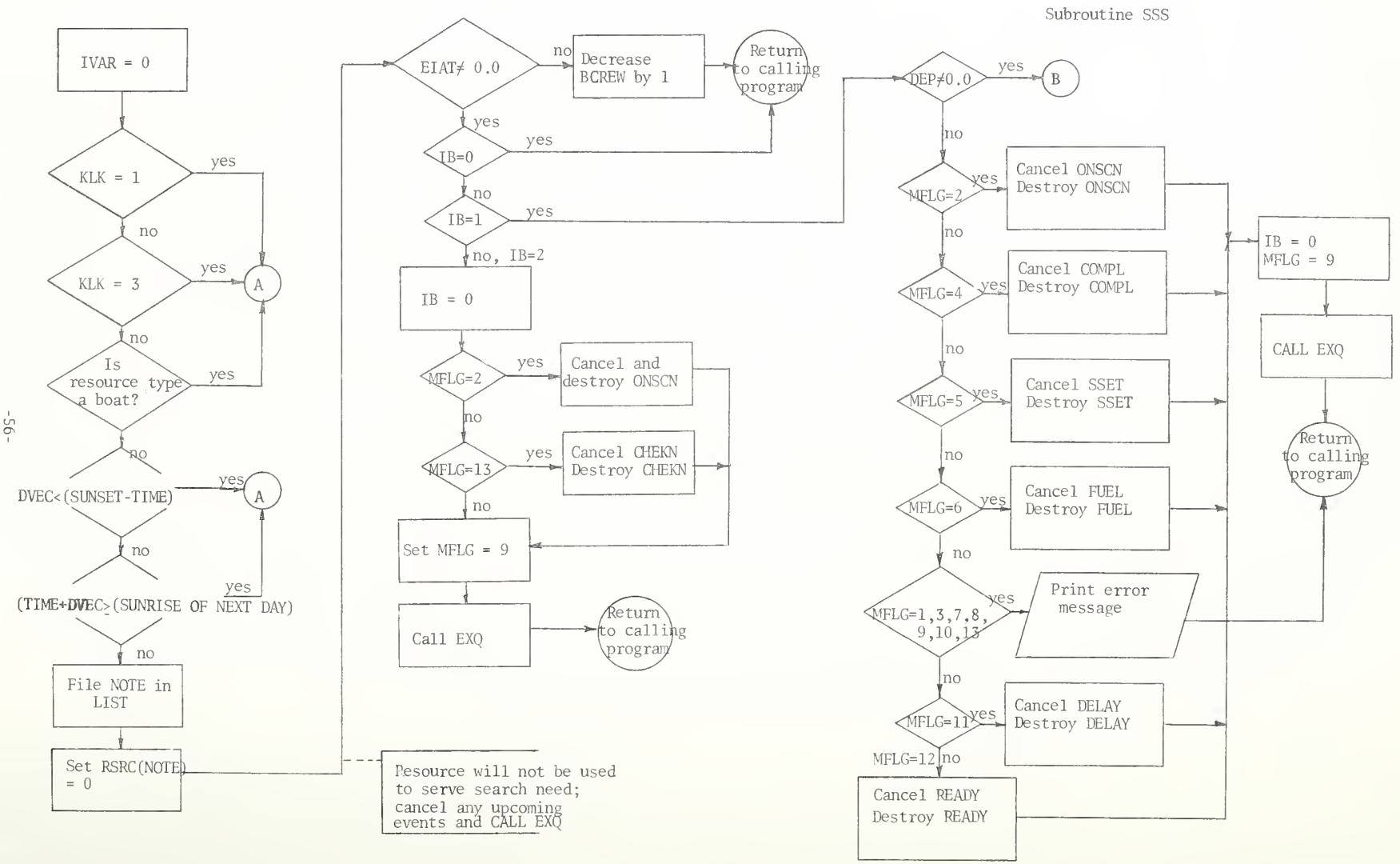

B
Remove first NOTE from SRHS (CASE)
Destroy NOTE
SRHS(CASE) is not empty
yes
no
IJK=1
yes
ITOL(CASE)=3
Increase FAIL1(STATN(CASE)) by 1
File CASE in EXCS
Return to calling program
no
ITOL(CASE)=4
Increase FAIL2(STATN(CASE)) by 1

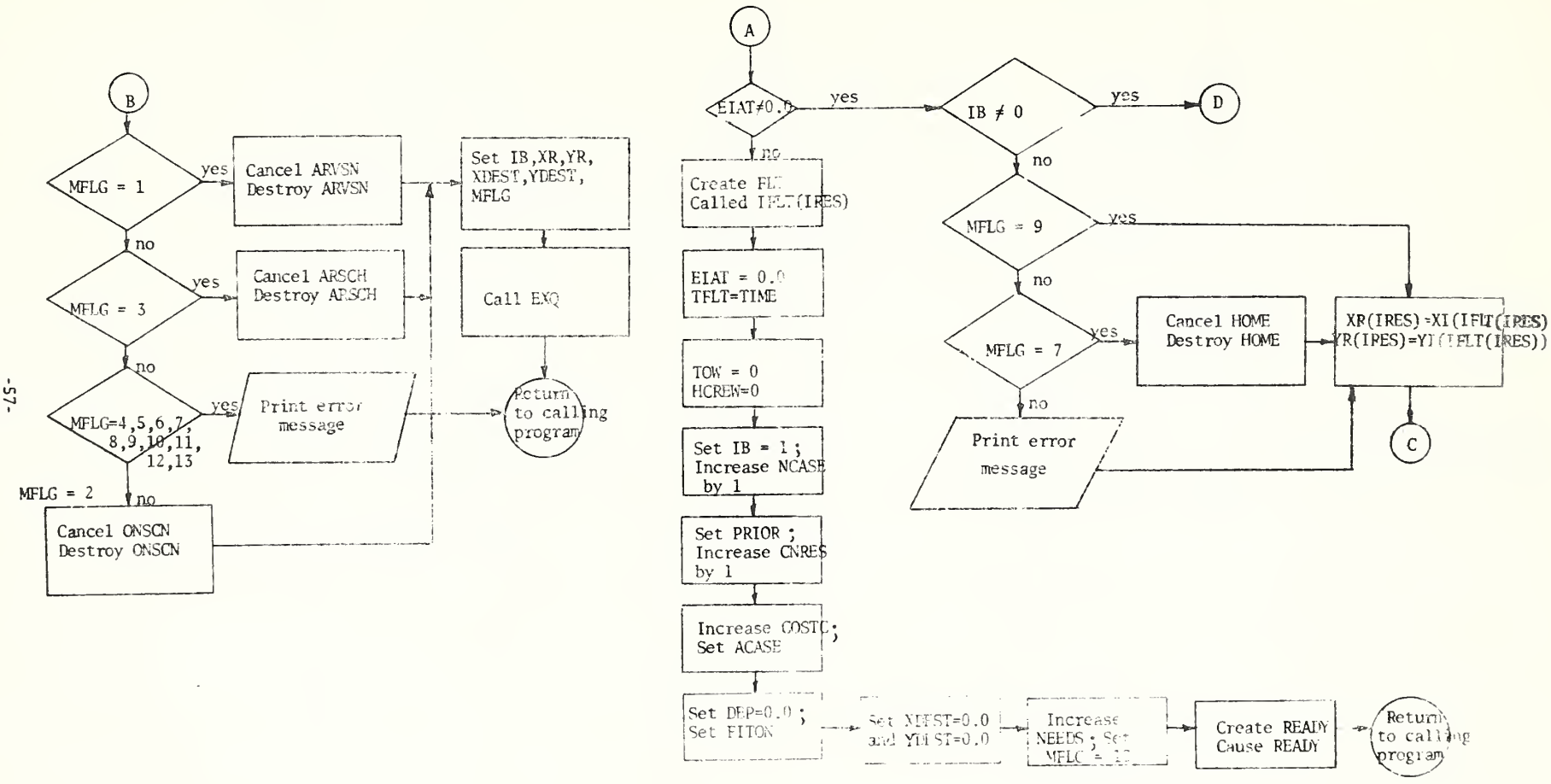

SUNSET<TIME<SUNRISE
-XRX
no
CALL SASS
RAP = 4 ?
yes
Change resource assignment policy to 2
no
RAP = 5 ?
yes
Change resource assignment policy to 6
no
File NOTE in LIST
Return to MAINXX
CALL RESAP
Was a resource found to serve the search need?
yes
Set SASG; Set RSRC
Is this the first resource to be assigned
no
CALL SSS
Return to MAINXX
SET SFLAG = 1
no
Are any other search needs of CASE in COUE
no
SET TINQ(CASE)
yes
Increase NQUE; Set REA; Set SIGNL
File NOTE in COUE
Return to MAINXX

Subroutine SASS

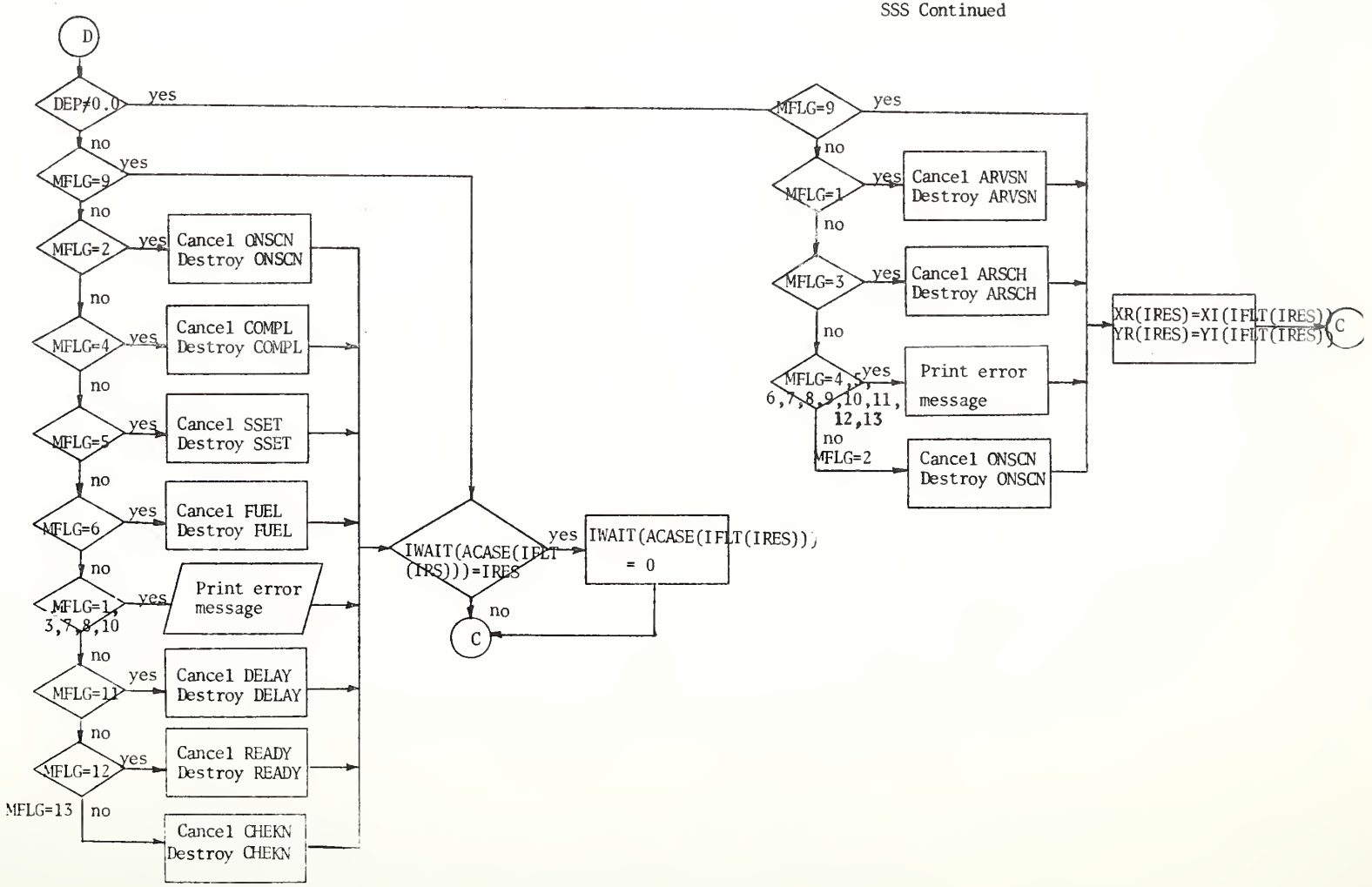


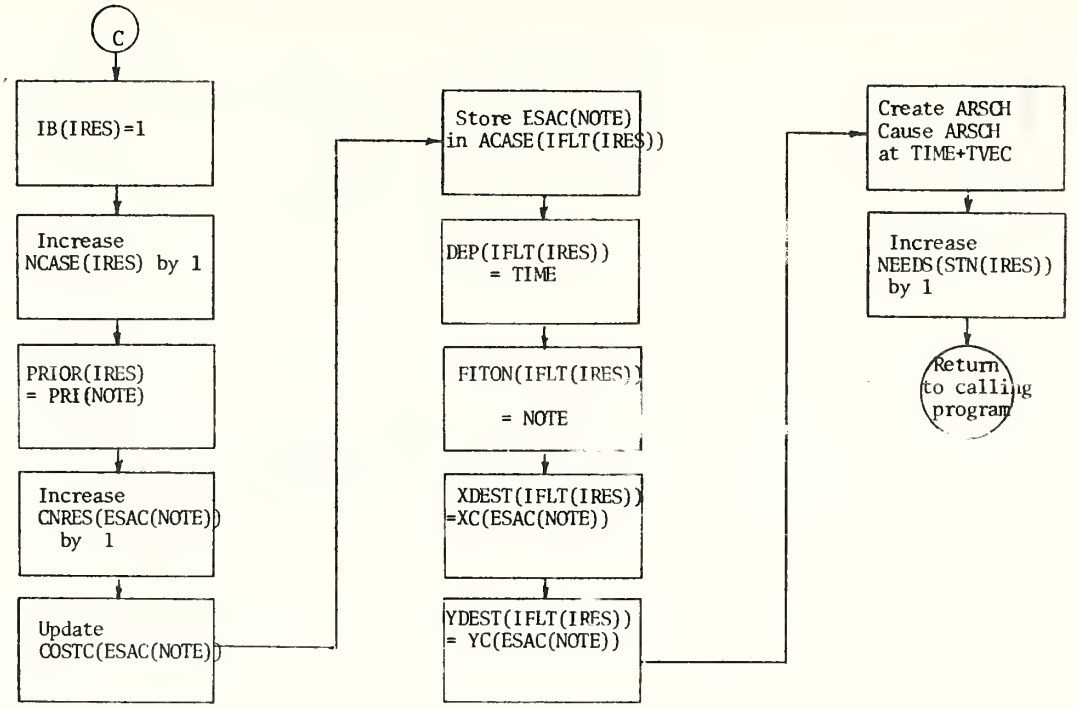

A
S2S(CASE) = 2
yes
PERC = 1.0
no
SDAY(NOTE) = 1
yes
PERC=PDC1
no
SDAY(NOTE) >1
yes
PERC=PDC2
no
Is RQUE empty
yes
no
Remove first IRS from RQUE
PR(IRS)>0.0
yes
PR(IRS)≥PERC
yes
File IRS in CSET
no
File IRS in PSET
no
Calculate X
X is the time to vector from the resource's home station to the location of the CASE
(END(TYPE(IRS)) - (2.0*X)) ≤ 0.0
yes
no
File IRS in TSET
CSET is empty
no
Remove first IRS from CSET
File IRS in RQUE
yes
PSET is empty
no
Remove first IRS from PSET
File IRS in RQUE
yes
TSET is empty
no
Remove first IRS from TSET
File IRS in RQUE
yes
RQUE is empty
yes
IDC = 3
yes
Return to calling program


Subroutine SSS
IVAR = 0
KLK = 1
yes
no
KLK = 3
yes
A
no
Is resource type a boat?
yes
A
no
DVEC<(SUNSET-TIME)
yes
A
no
(TIME+DVEC>(SUNRISE OF NEXT DAY)
yes
no
File NOTE in LIST
Set RSRC(NOTE) = 0
Resource will not be used to serve search need; cancel any upcoming events and CALL EXQ
EIAT≠ 0.0
no
Decrease BCREW by 1
Return to calling program
yes
IB=0
yes
no
IB=1
yes
no, IB=2
IB = 0
MFLG=2
yes
Cancel and destroy ONSCN
no
MFLG=13
yes
Cancel CHEKN Destroy CHEKN
no
Set MFLG = 9
Call EXQ
Return to calling program
DEP≠0.0
yes
B
no
MFLG=2
yes
Cancel ONSCN Destroy ONSCN
no
MFLG=4
yes
Cancel COMPL Destroy COMPL
no
MFLG=5
yes
Cancel SSET Destroy SSET
no
MFLG=6
yes
Cancel FUEL Destroy FUEL
no
MFLG=1,3,7,8, 9,10,13
yes
Print error message
no
MFLG=11
yes
Cancel DELAY Destroy DELAY
MFLG=12
no
Cancel READY Destroy READY
IB = 0
MFLG = 9
CALL EXQ
Return to calling program

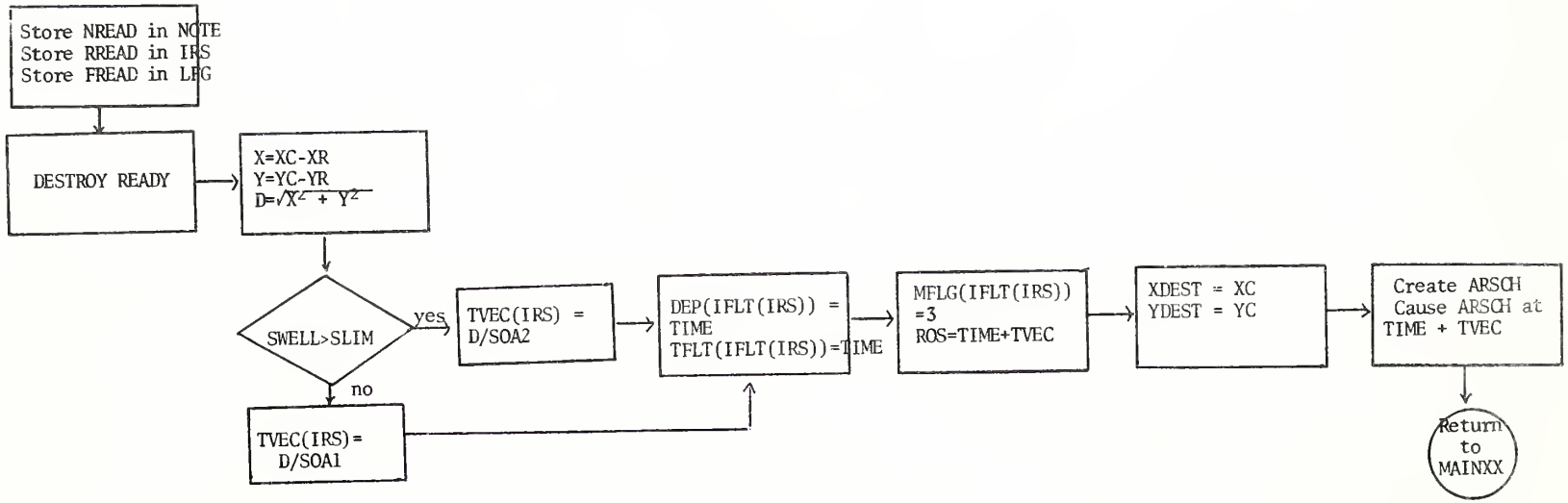

A
B
C
D
EIAT≠0.0
yes
no
IB ≠ 0
yes
no
MFLG = 9
yes
no
MFLG = 7
yes
no
Cancel HOME
Destroy HOME
XR(IRES)=XI(IFLT(IRES)
YR(IRES)=YI(IFLT(IRES))
Print error
message
Create FLT
Called IFLT(IRES)
EIAT = 0.0
TFLT=TIME
TOW = 0
HCREW=0
Set IB = 1;
Increase NCASE
by 1
Set PRIOR ;
Increase CNRES
by 1
Increase COSTC;
Set ACASE
Set DEP=0.0 ;
Set FITON
Set XIFST=0.0
and YIFST=0.0
Increase
NEEDS ; Set
MFLG = 12
Create READY
Cause READY
Return
to calling
program
MFLG = 1
yes
Cancel ARVSN
Destroy ARVSN
Set IB,XR,YR,
XDEST,YDEST,
MFLG
MFLG = 3
yes
Cancel ARSCH
Destroy ARSCH
Call EXQ
MFLG=4,5,6,7,
8,9,10,11,
12,13
yes
Print error
message
Return
to calling
program
MFLG = 2
no
Cancel ONSCN
Destroy ONSCN

SSS Continued

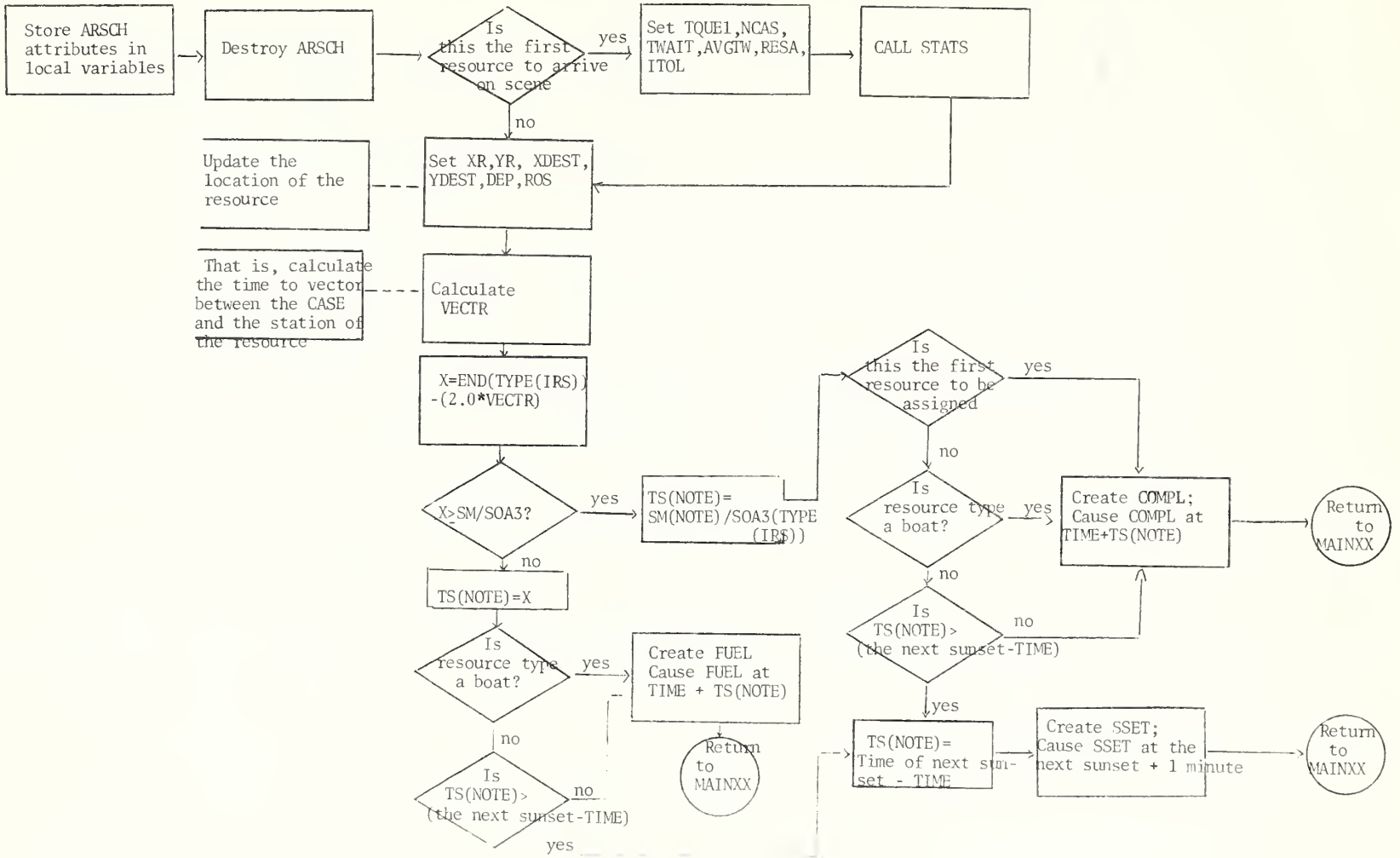


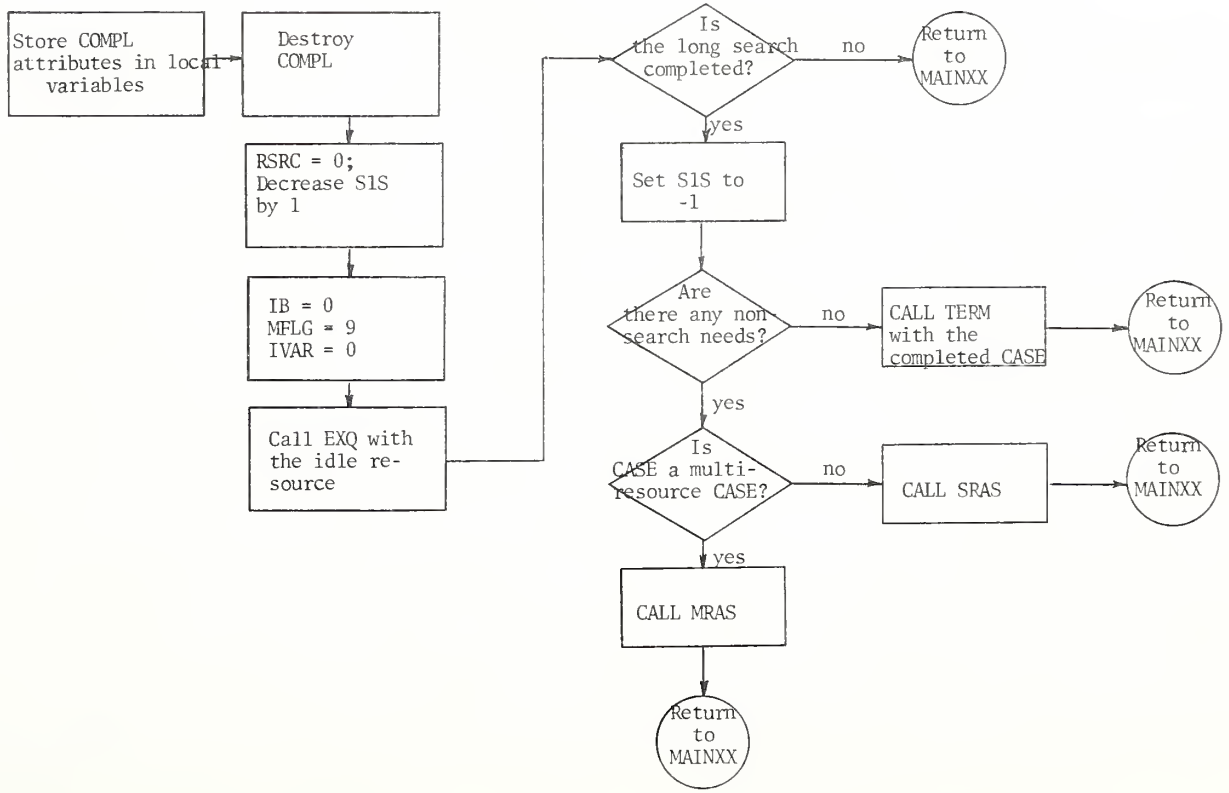

C
IB(IRES)=1
Increase NCASE(IRES) by 1
PRIOR(IRES) = PRI(NOTE)
Increase CNRES(ESAC(NOTE)) by 1
Update COSTC(ESAC(NOTE))
Store ESAC(NOTE) in ACASE(IFLT(IRES))
DEP(IFLT(IRES)) = TIME
FITON(IFLT(IRES)) = NOTE
XDEST(IFLT(IRES)) =XC(ESAC(NOTE))
YDEST(IFLT(IRES)) = YC(ESAC(NOTE))
Create ARSCH Cause ARSCH at TIME+TVEC
Increase NEEDS(STN(IRES)) by 1
Return to calling program

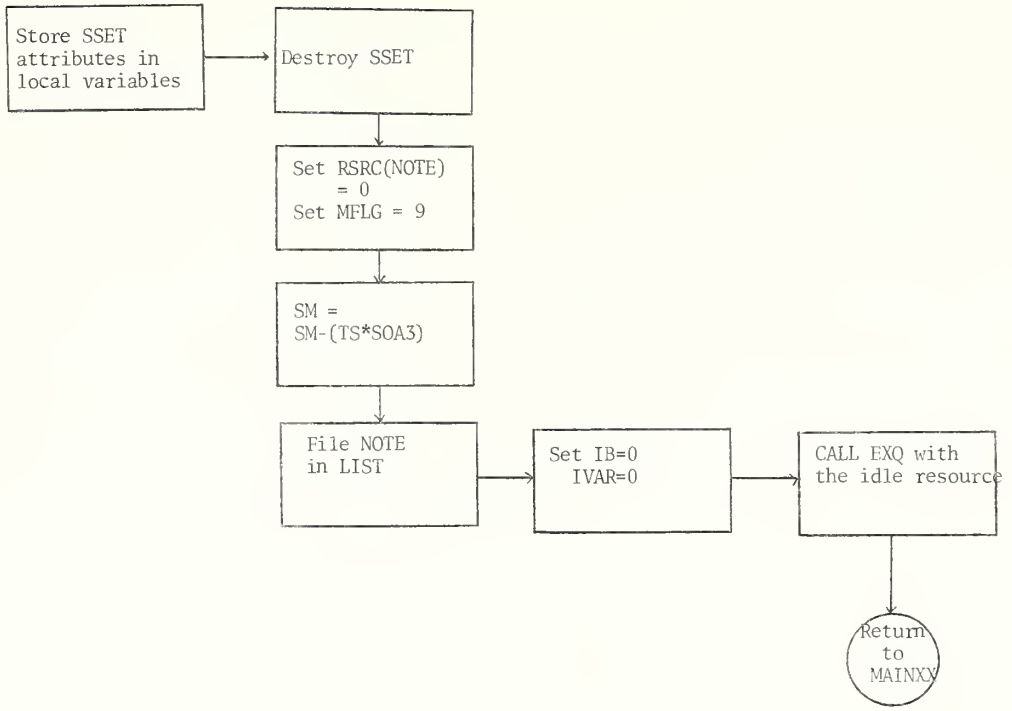
Subroutine READY



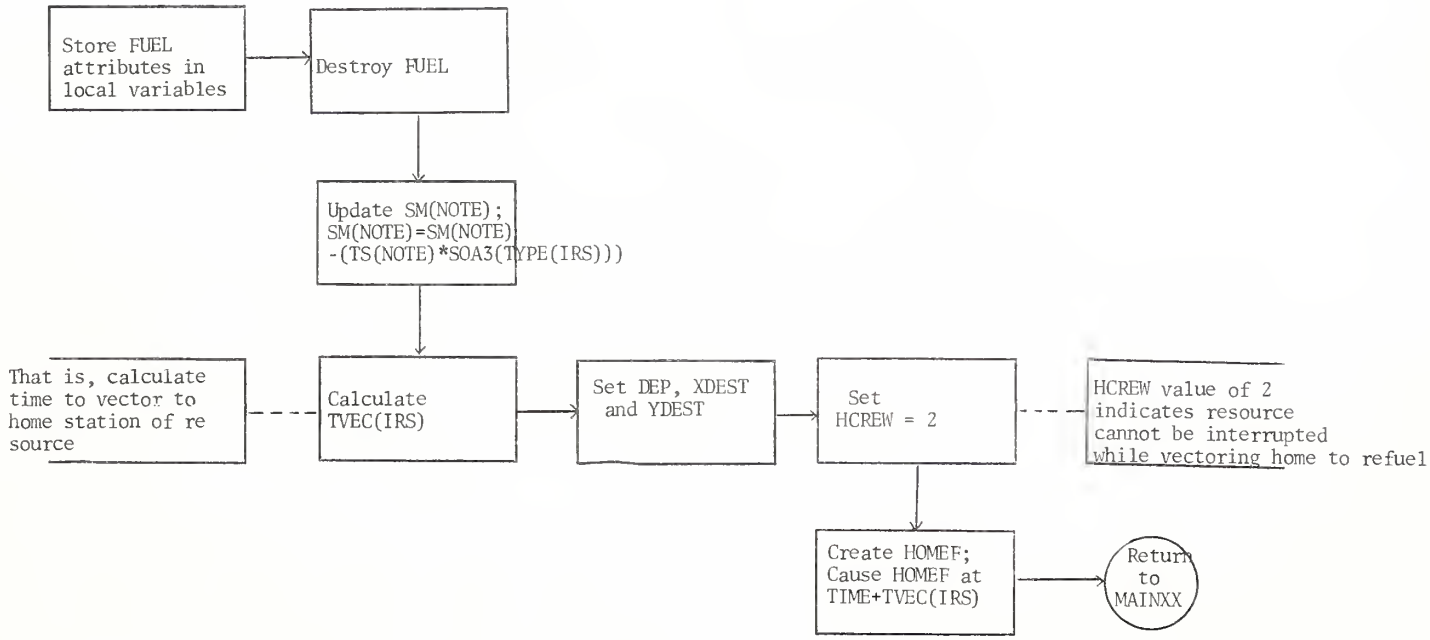
Endogenous Event ARSCH

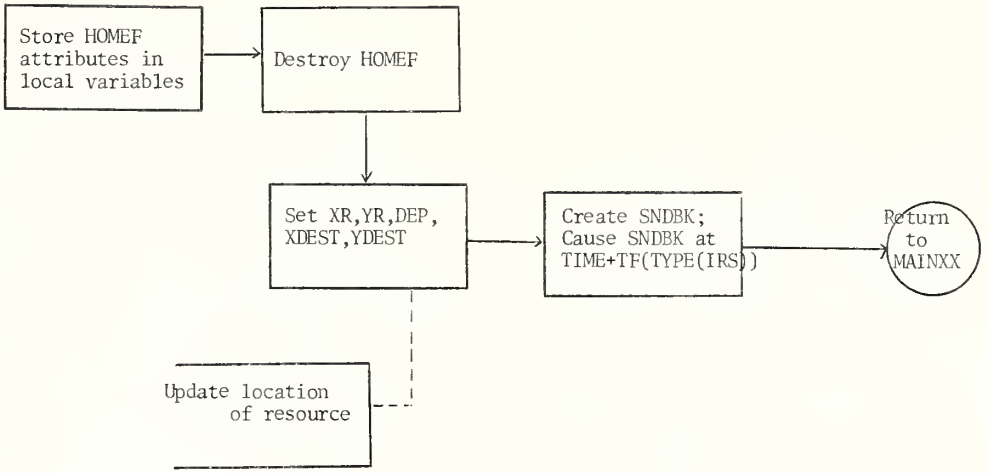


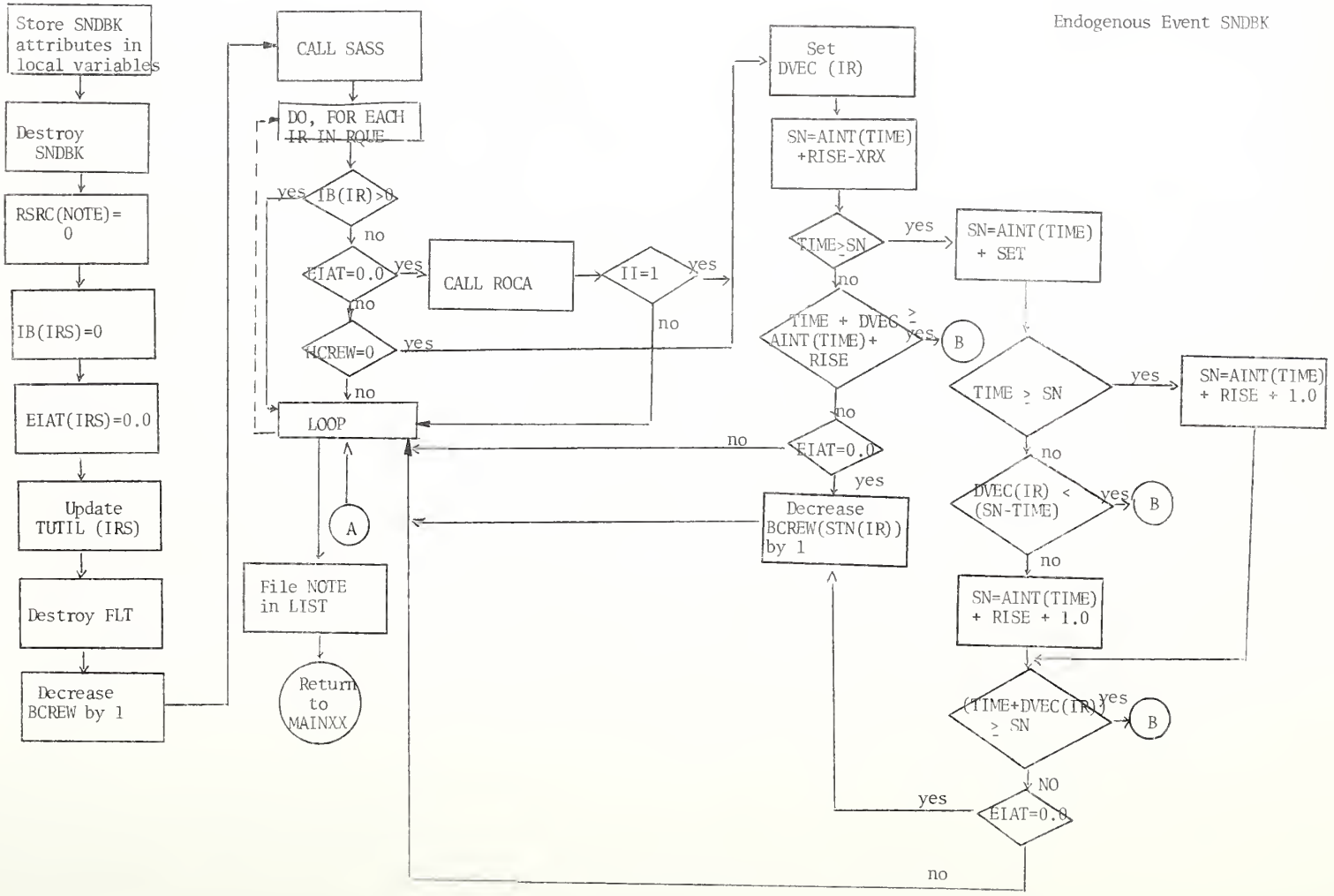

Store COMPL attributes in local variables
Destroy COMPL
RSRC = 0; Decrease S1S by 1
IB = 0
MFLG = 9
IVAR = 0
Call EXQ with the idle resource
Is the long search completed?
no
Return to MAINXX
yes
Set S1S to -1
Are there any non-search needs?
no
CALL TERM with the completed CASE
Return to MAINXX
yes
Is CASE a multi-resource CASE?
no
CALL SRAS
Return to MAINXX
yes
CALL MRAS
Return to MAINXX

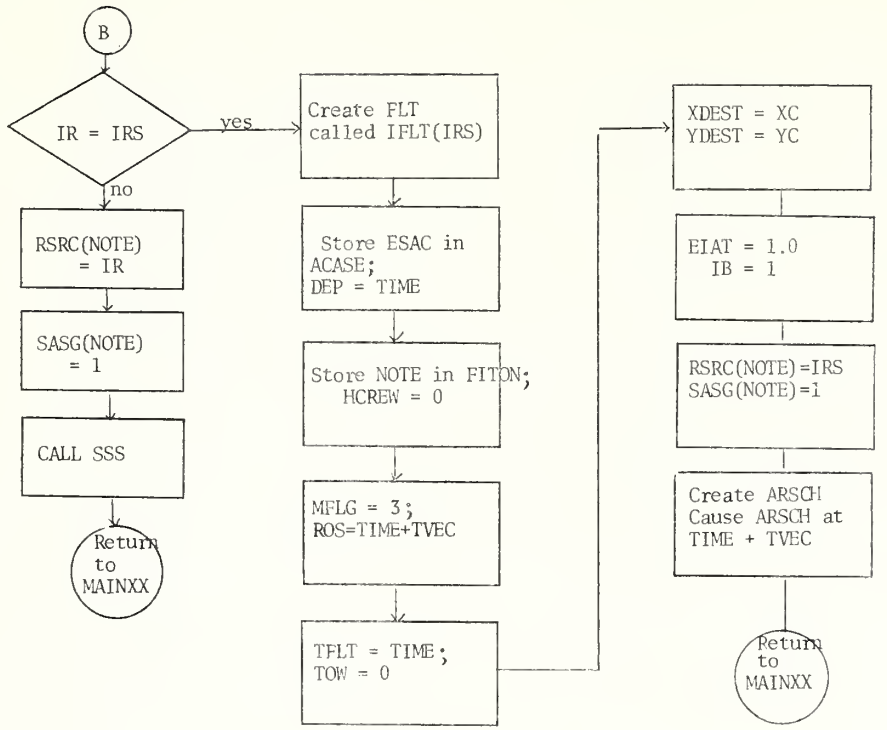

Store SSET attributes in local variables
Destroy SSET
Set RSRC(NOTE) = 0
Set MFLG = 9
SM = SM-(TS*SOA3)
File NOTE in LIST
Set IB=0 IVAR=0
CALL EXQ with the idle resource
Return to MAINXX

Endogenous Event FUEL

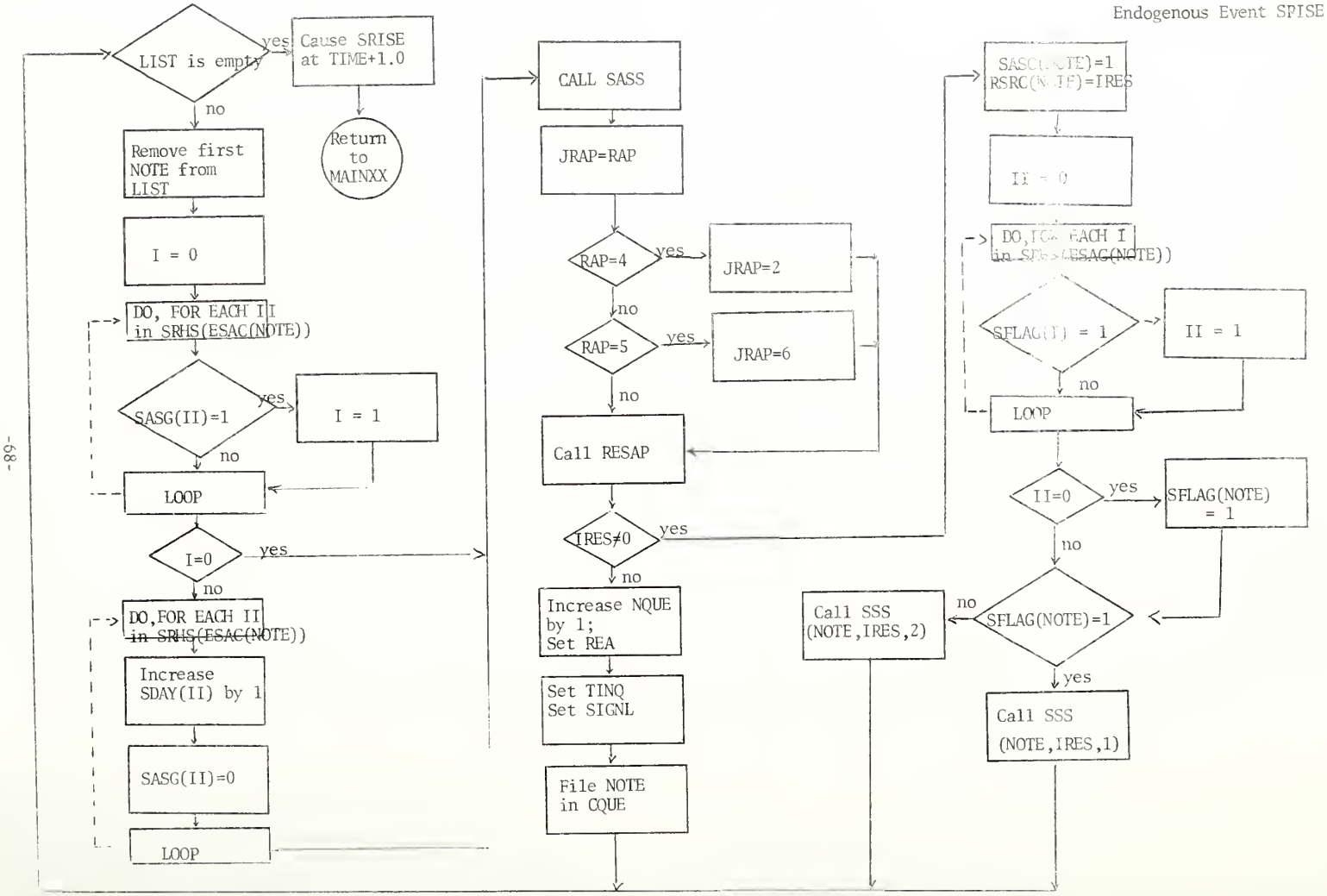

Store FUEL attributes in local variables
Destroy FUEL
Update SM(NOTE); SM(NOTE)=SM(NOTE) -(TS(NOTE)*SOA3(TYPE(IRS)))
That is, calculate time to vector to home station of re source
Calculate TVEC(IRS)
Set DEP, XDEST and YDEST
Set HCREW = 2
HCREW value of 2 indicates resource cannot be interrupted while vectoring home to refuel
Create HOMEF; Cause HOMEF at TIME+TVEC(IRS)
Return to MAINXX


Store HOMEF attributes in local variables
Destroy HOMEF
Set XR,YR,DEP, XDEST,YDEST
Create SNDBK; Cause SNDBK at TIME+TF(TYPE(IRS))
Return to MAINXX
Update location of resource

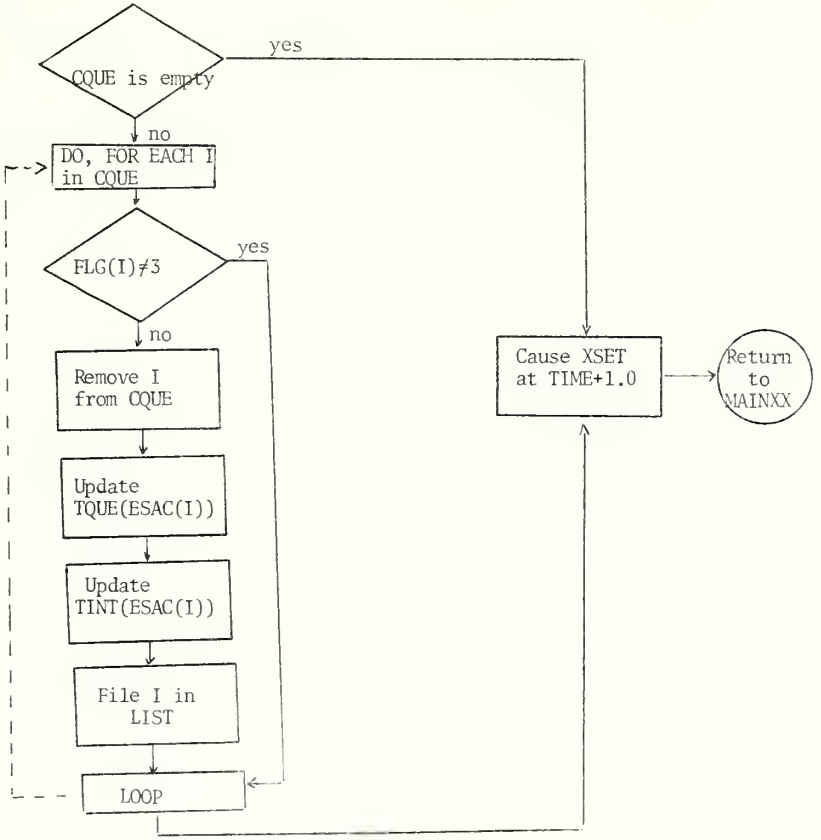

Endogenous Event SNDBK
Store SNDBK attributes in local variables
Destroy SNDBK
RSRC(NOTE)=0
IB(IRS)=0
EIAT(IRS)=0.0
Update TUTIL (IRS)
Destroy FLT
Decrease BCREW by 1
CALL SASS
DO, FOR EACH IR IN ROUE
IB(IR)>0
yes
no
EIAT=0.0
yes
no
CALL ROCA
II=1
yes
no
NCREW=0
yes
no
LOOP
A
File NOTE in LIST
Return to MAINXX
Set DVEC (IR)
SN=AINT(TIME) +RISE-XRX
TIME≥SN
yes
no
SN=AINT(TIME) + SET
TIME + DVEC ≥ AINT(TIME)+ RISE
yes
no
B
TIME ≥ SN
yes
no
SN=AINT(TIME) + RISE + 1.0
EIAT=0.0
yes
no
Decrease BCREW(STN(IR)) by 1
DVEC(IR) < (SN-TIME)
yes
no
B
SN=AINT(TIME) + RISE + 1.0
(TIME+DVEC(IR)) ≥ SN
yes
NO
B
EIAT=0.0
yes
no


B
IR = IRS
yes
no
Create FLT called IFLT(IRS)
RSRC(NOTE) = IR
SASG(NOTE) = 1
CALL SSS
Return to MAINXX
Store ESAC in ACASE; DEP = TIME
Store NOTE in FITON; HCREW = 0
MFLG = 3; ROS=TIME+TVEC
TFLT = TIME; TOW = 0
XDEST = XC
YDEST = YC
EIAT = 1.0
IB = 1
RSRC(NOTE)=IRS
SASG(NOTE)=1
Create ARSCH Cause ARSCH at TIME + TVEC
Return to MAINXX

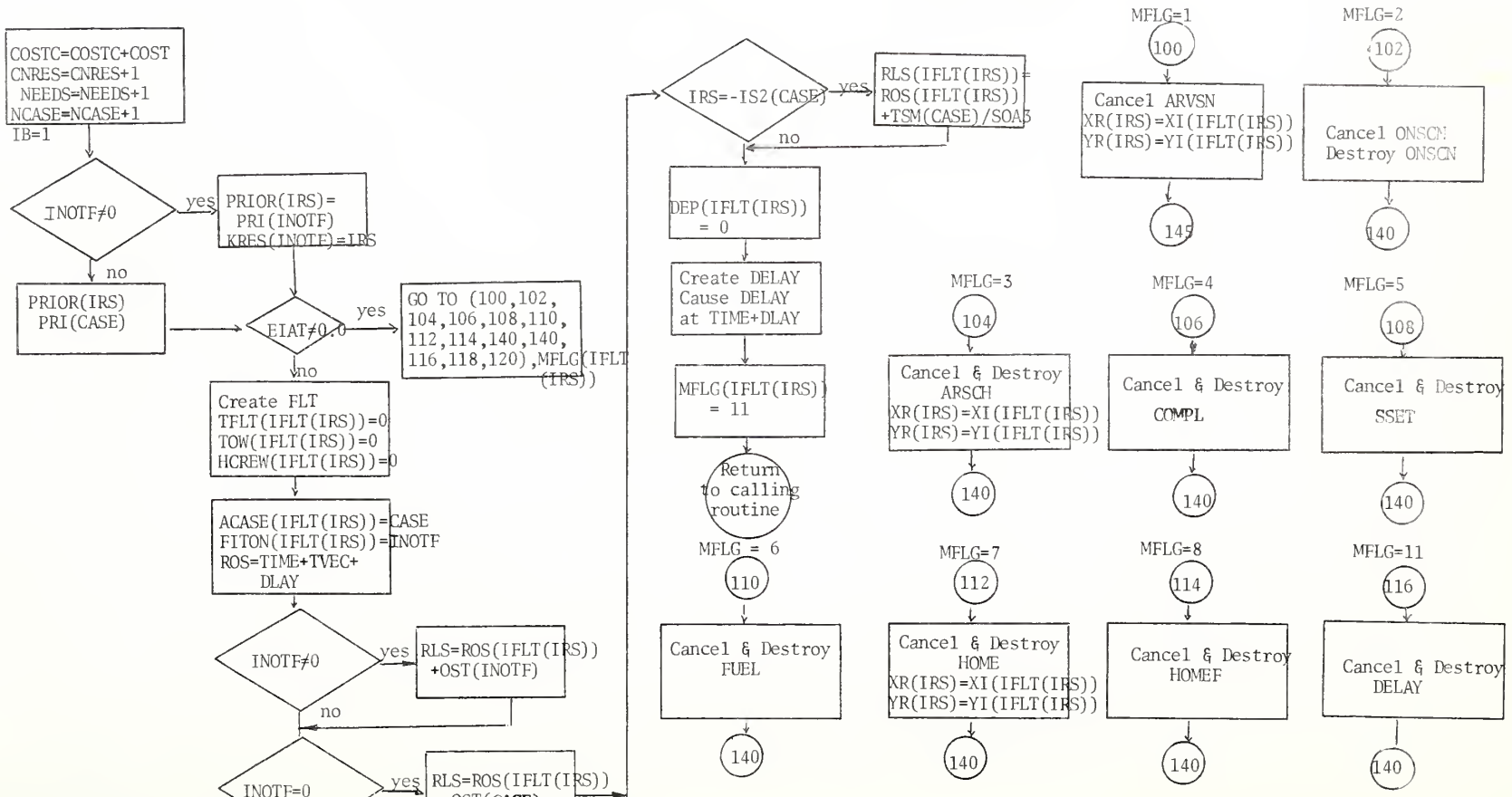
Endogenous Event SRISE

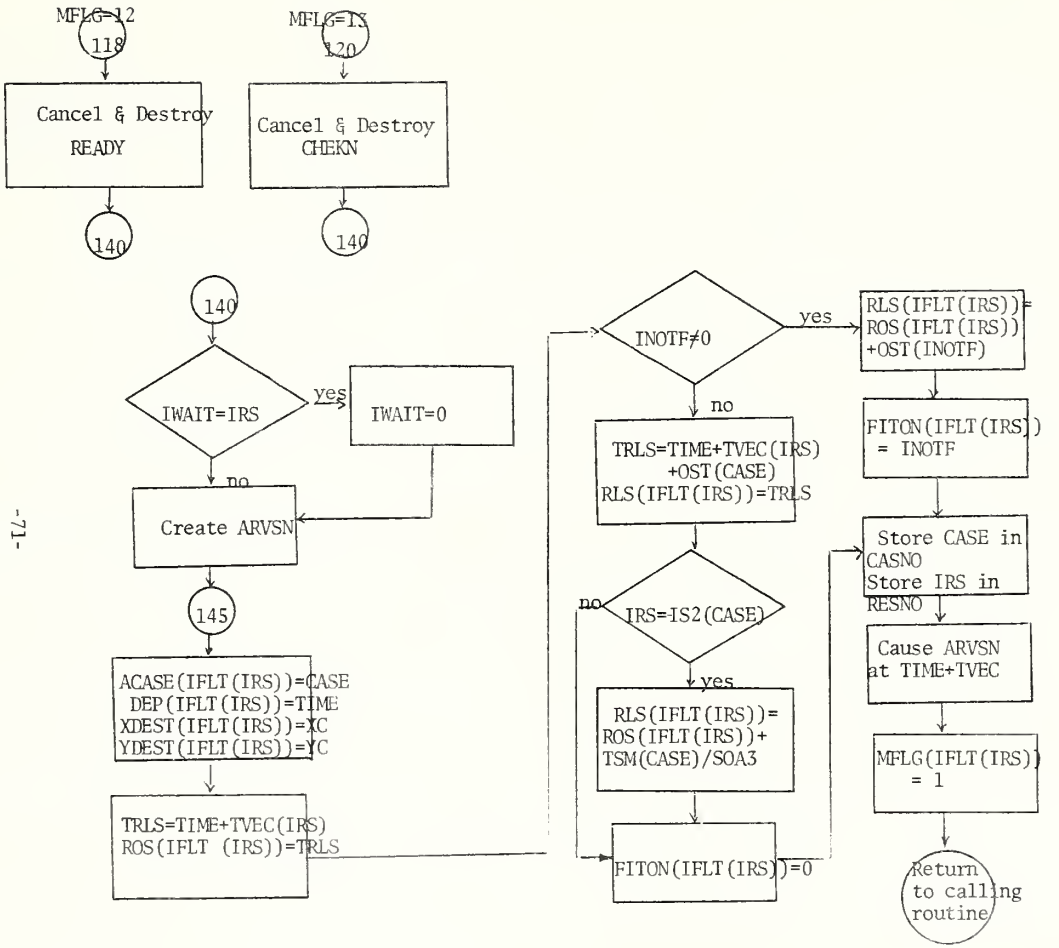


Endogenous Event XSET

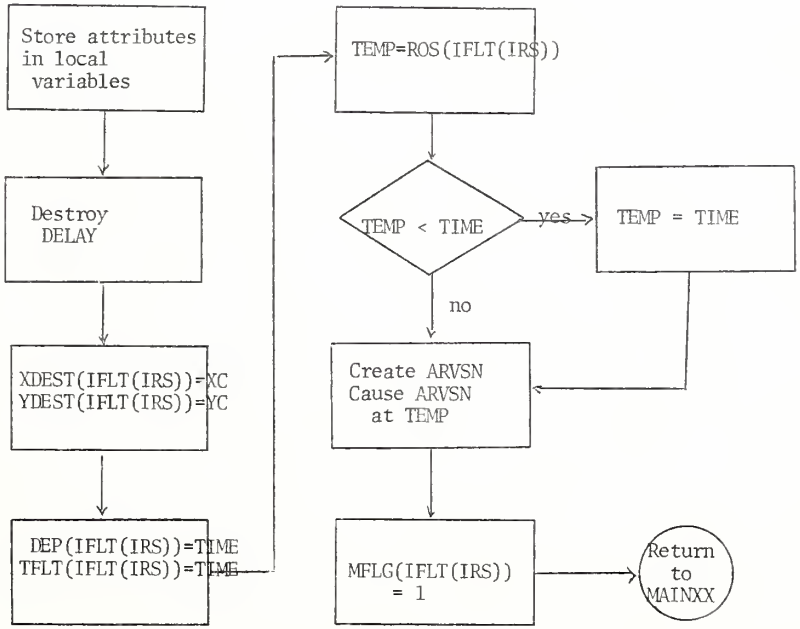


Subroutine SERVE

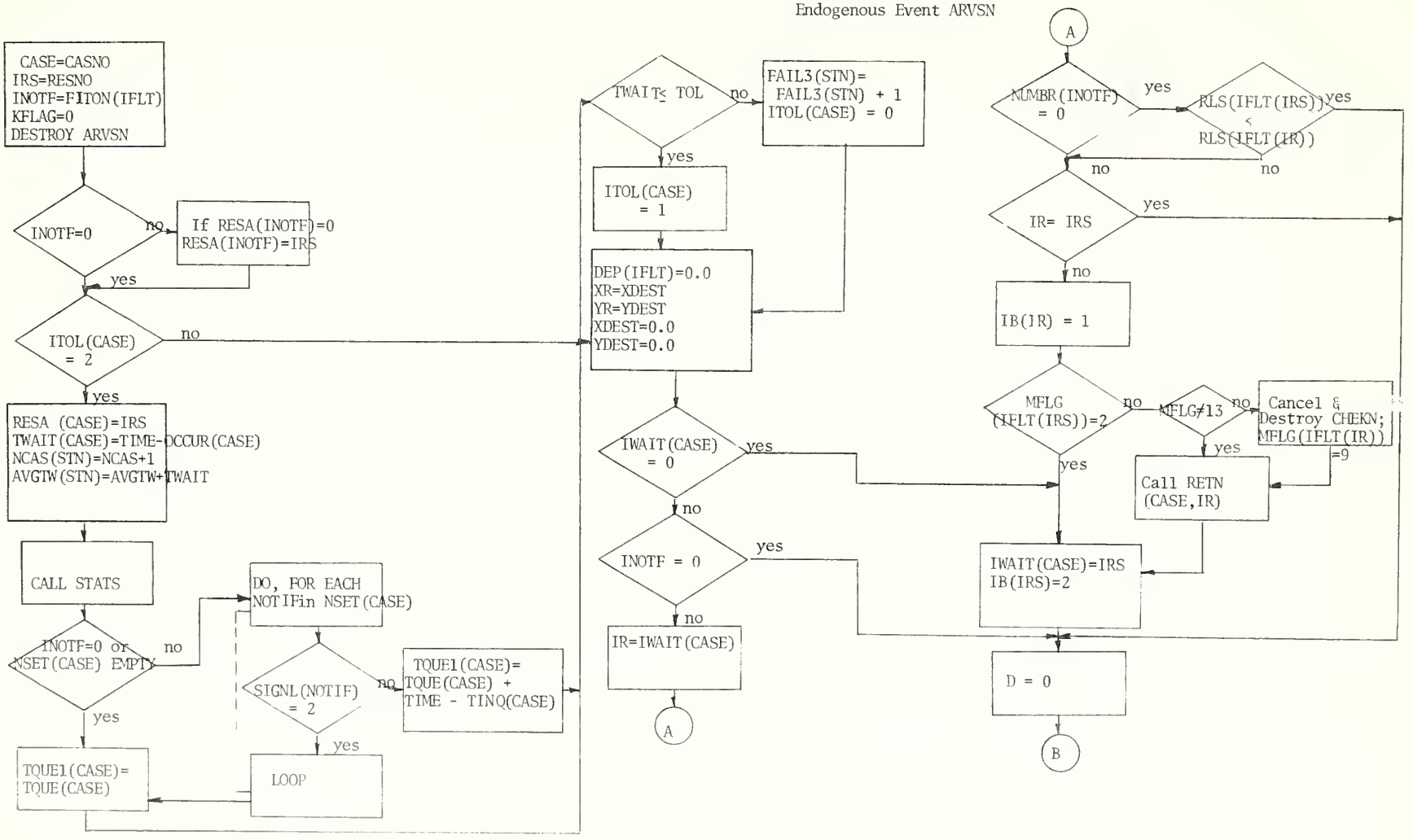



MFLG=12
118
Cancel & Destroy
READY
140
MFLG=13
120
Cancel & Destroy
CHEKN
140
140
IWAIT=IRS
yes
IWAIT=0
no
Create ARVSN
145
ACASE(IFLT(IRS))=CASE
DEP(IFLT(IRS))=TIME
XDEST(IFLT(IRS))=XC
YDEST(IFLT(IRS))=YC
TRLS=TIME+TVEC(IRS)
ROS(IFLT (IRS))=TRLS
INOTF≠0
yes
no
TRLS=TIME+TVEC(IRS)
+OST(CASE)
RLS(IFLT(IRS))=TRLS
IRS=IS2(CASE)
no
yes
RLS(IFLT(IRS))=
ROS(IFLT(IRS))+
TSM(CASE)/SOA3
FITON(IFLT(IRS))=0
RLS(IFLT(IRS))=
ROS(IFLT(IRS))
+OST(INOTF)
FITON(IFLT(IRS))
= INOTF
Store CASE in
CASNO
Store IRS in
RESNO
Cause ARVSN
at TIME+TVEC
MFLG(IFLT(IRS))
= 1
Return
to calling
routine

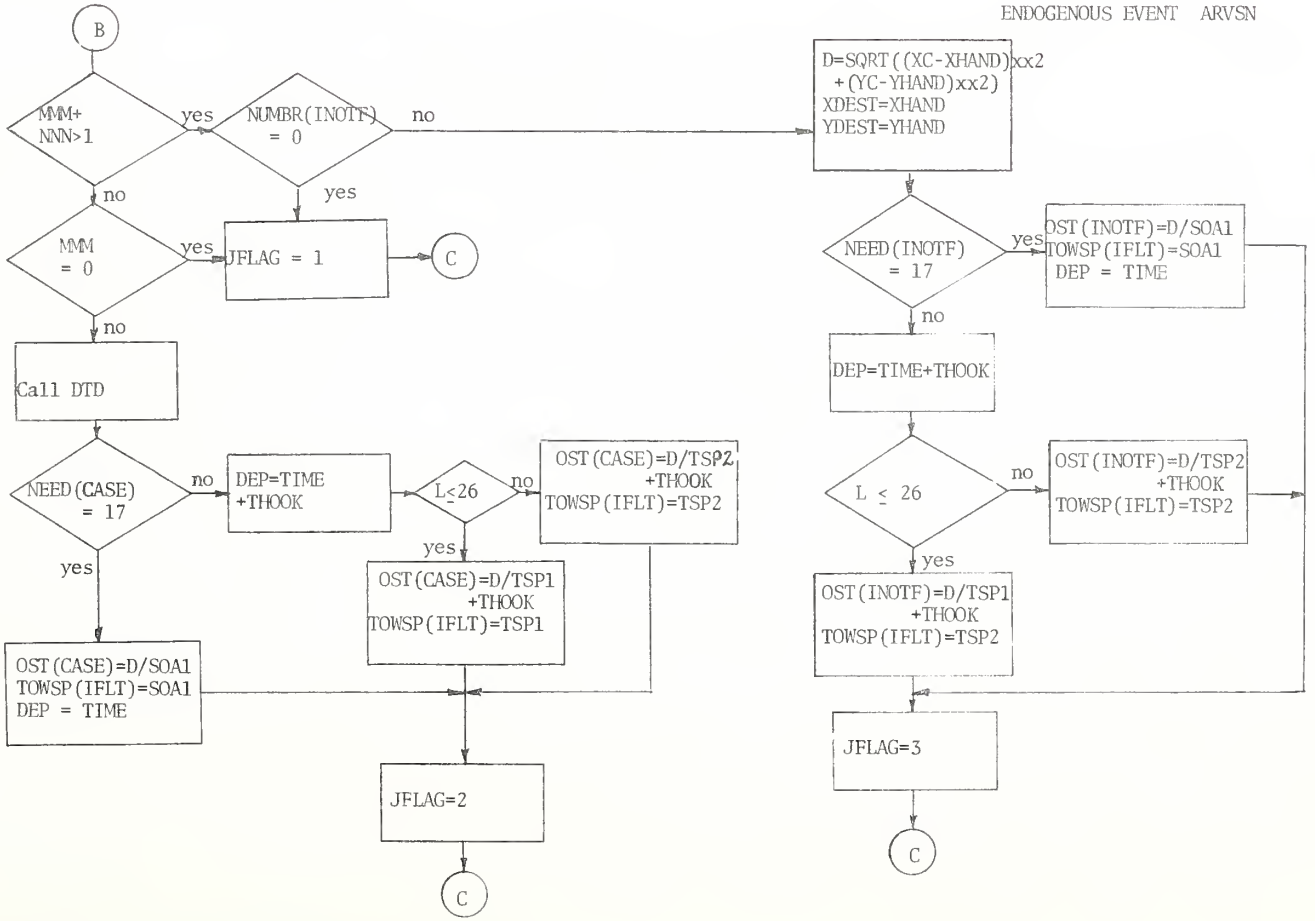
ENDOGENOUS EVENT DELAY

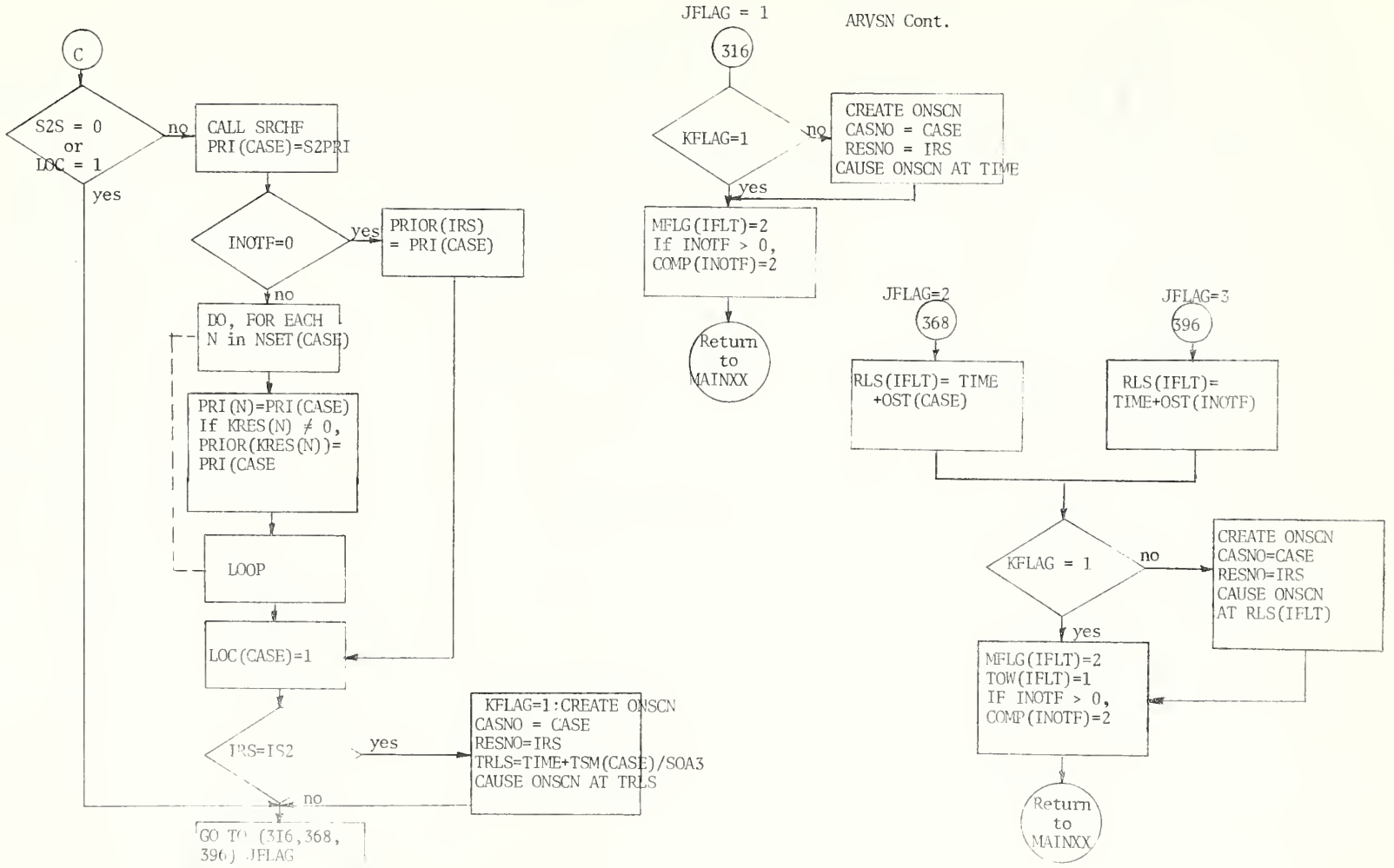


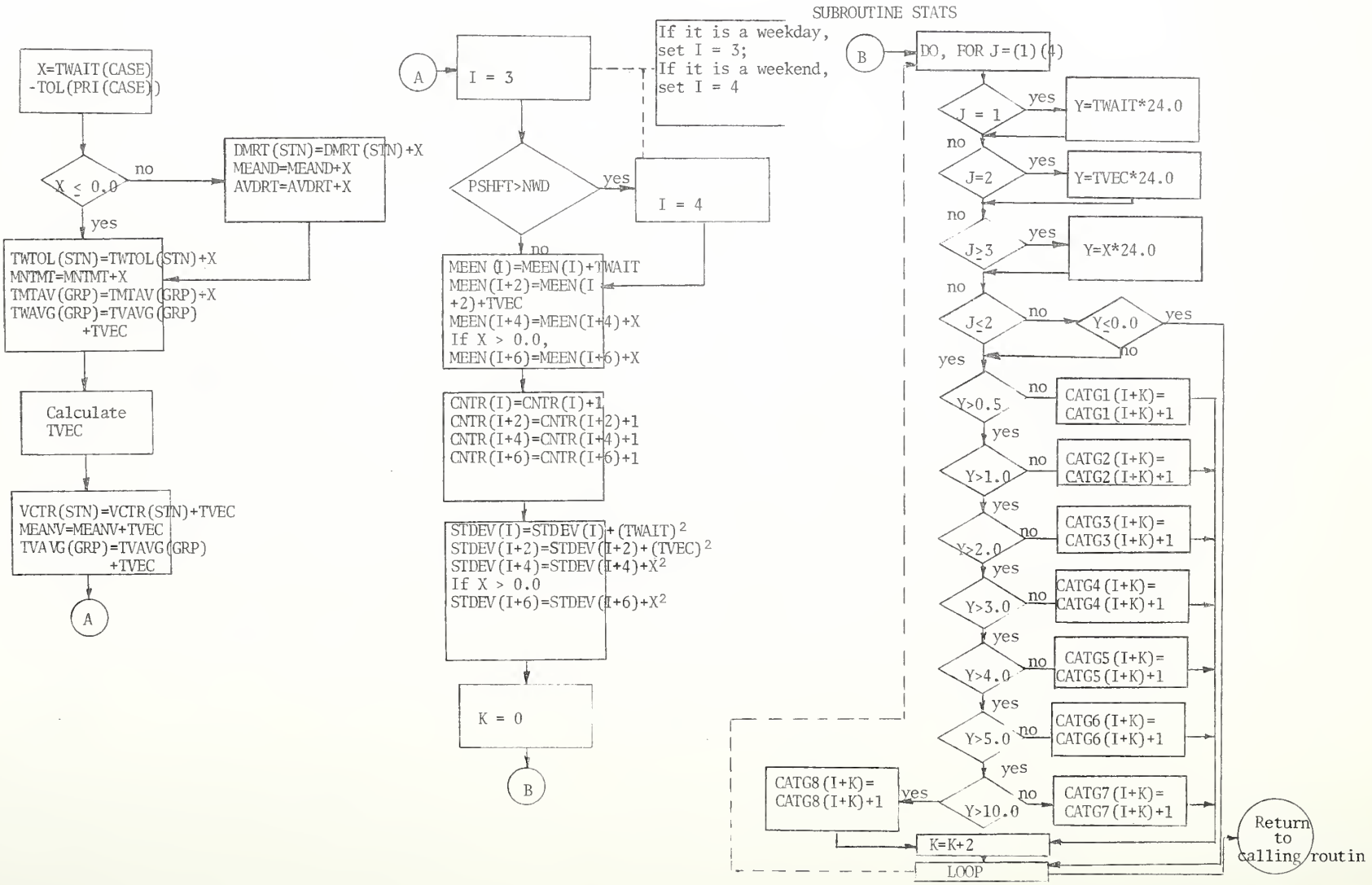
Endogenous Event ARVSN



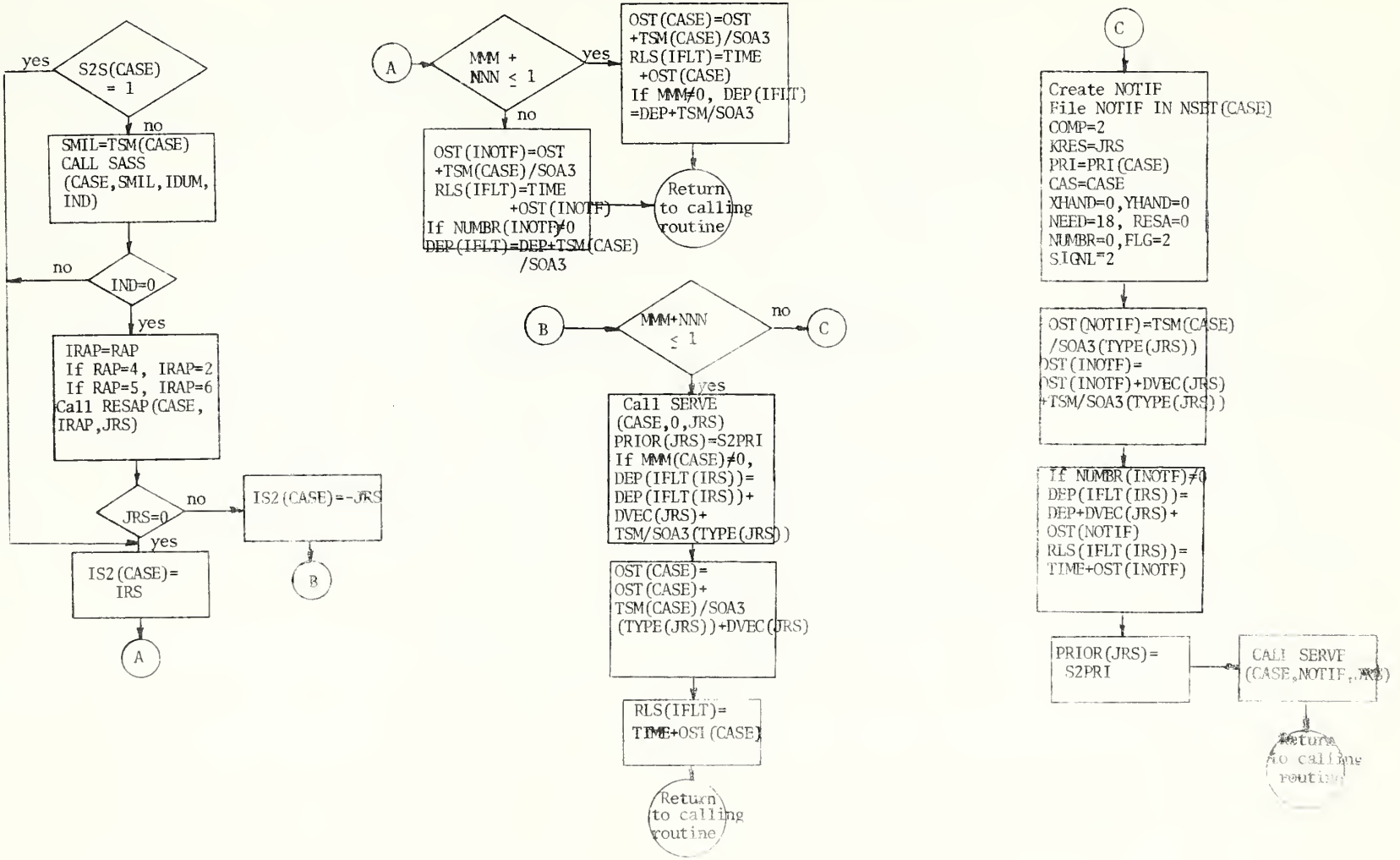
ENDOGENOUS EVENT ARVSN



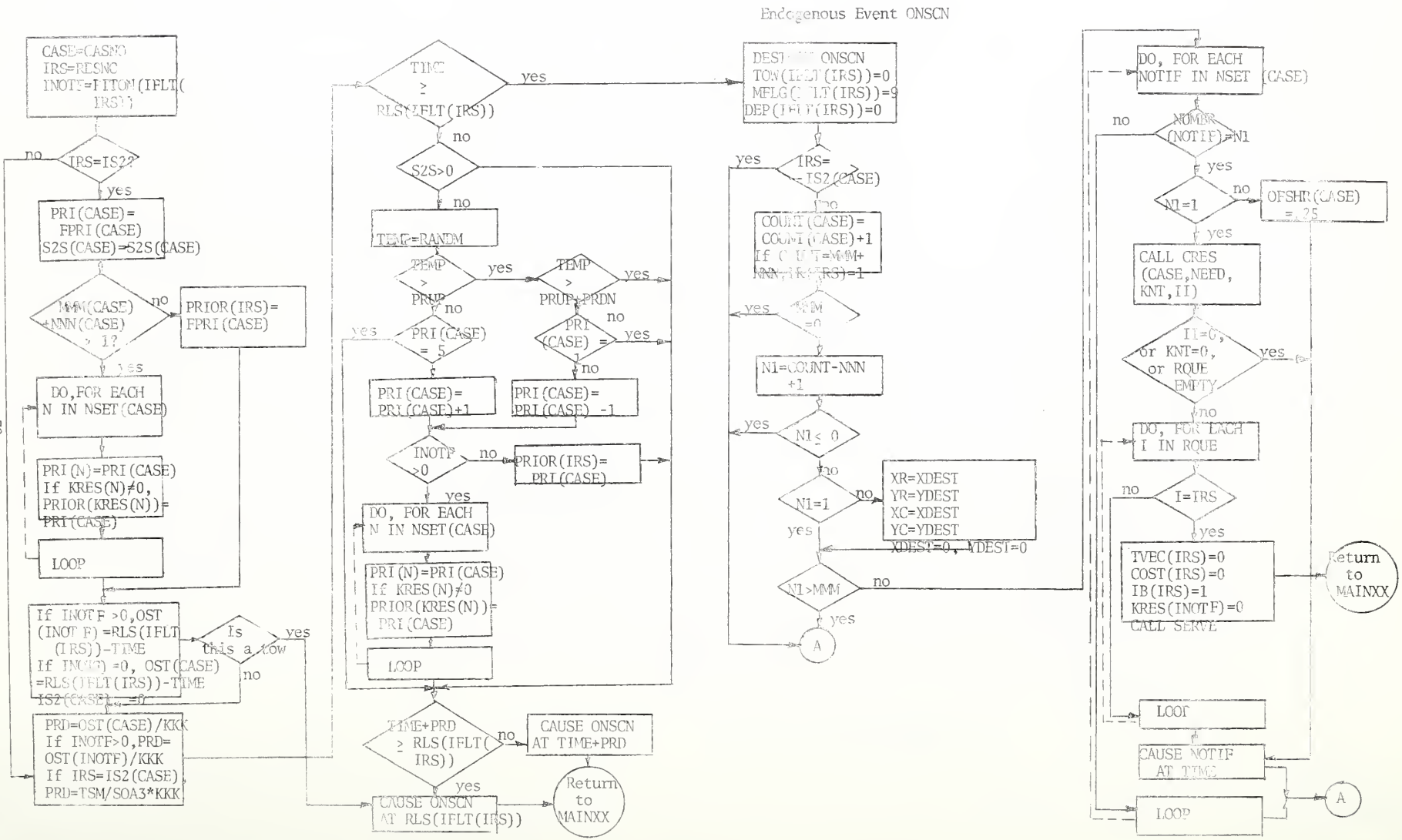

C
S2S = 0
or
LOC = 1
no
CALL SRCHF
PRI(CASE)=S2PRI
yes
INOTF=0
yes
PRIOR(IRS)
= PRI(CASE)
no
DO, FOR EACH
N in NSET(CASE)
PRI(N)=PRI(CASE)
If KRES(N) ≠ 0,
PRIOR(KRES(N))=
PRI(CASE
LOOP
LOC(CASE)=1
IRS=IS2
yes
KFLAG=1:CREATE ONSCN
CASNO = CASE
RESNO=IRS
TRLS=TIME+TSM(CASE)/SOA3
CAUSE ONSCN AT TRLS
no
GO TO (316,368,
396) JFLAG
JFLAG = 1
316
KFLAG=1
no
CREATE ONSCN
CASNO = CASE
RESNO = IRS
CAUSE ONSCN AT TIME
yes
MFLG(IFLT)=2
If INOTF > 0,
COMP(INOTF)=2
Return
to
MAINXX
ARVSN Cont.
JFLAG=2
368
RLS(IFLT)= TIME
+OST(CASE)
JFLAG=3
396
RLS(IFLT)=
TIME+OST(INOTF)
KFLAG = 1
no
CREATE ONSCN
CASNO=CASE
RESNO=IRS
CAUSE ONSCN
AT RLS(IFLT)
yes
MFLG(IFLT)=2
TOW(IFLT)=1
IF INOTF > 0,
COMP(INOTF)=2
Return
to
MAINXX

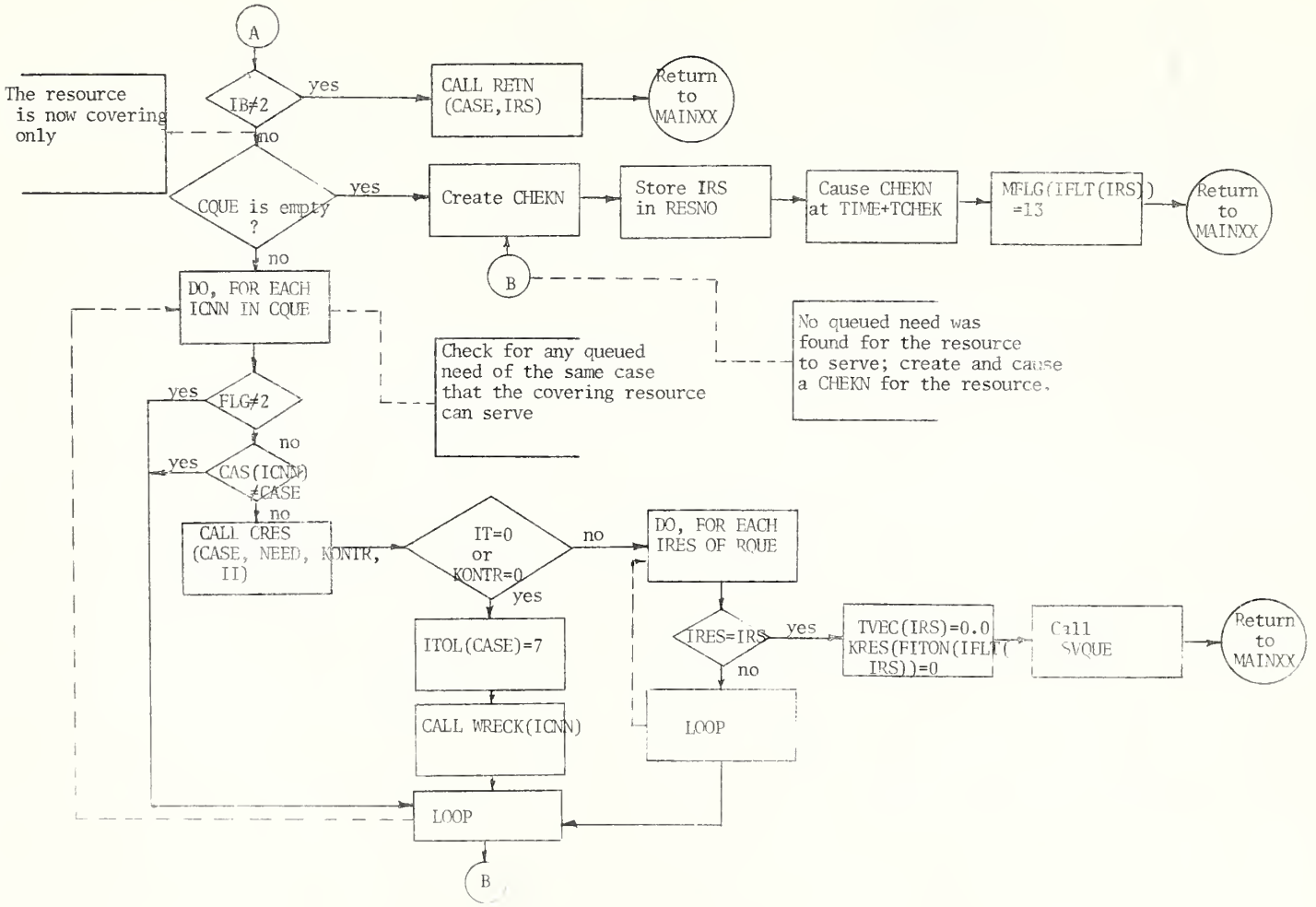

SUBROUTINE STATS
X=TWAIT(CASE)
-TOL(PRI(CASE))
X ≤ 0.0
no
yes
DMRT(STN)=DMRT(STN)+X
MEAND=MEAND+X
AVDRT=AVDRT+X
TWTOL(STN)=TWTOL(STN)+X
MNTMT=MNTMT+X
TMTAV(GRP)=TMTAV(GRP)+X
TWAVG(GRP)=TVAVG(GRP)
+TVEC
Calculate
TVEC
VCTR(STN)=VCTR(STN)+TVEC
MEANV=MEANV+TVEC
TVAVG(GRP)=TVAVG(GRP)
+TVEC
A
A
I = 3
If it is a weekday,
set I = 3;
If it is a weekend,
set I = 4
PSHFT>NWD
yes
I = 4
no
MEEN(I)=MEEN(I)+TWAIT
MEEN(I+2)=MEEN(I
+2)+TVEC
MEEN(I+4)=MEEN(I+4)+X
If X > 0.0,
MEEN(I+6)=MEEN(I+6)+X
CNTR(I)=CNTR(I)+1
CNTR(I+2)=CNTR(I+2)+1
CNTR(I+4)=CNTR(I+4)+1
CNTR(I+6)=CNTR(I+6)+1
STDEV(I)=STDEV(I)+(TWAIT)2
STDEV(I+2)=STDEV(I+2)+(TVEC)2
STDEV(I+4)=STDEV(I+4)+X2
If X > 0.0
STDEV(I+6)=STDEV(I+6)+X2
K = 0
B
B
DO, FOR J=(1)(4)
J = 1
yes
Y=TWAIT*24.0
no
J=2
yes
Y=TVEC*24.0
no
J≥3
yes
Y=X*24.0
no
J≤2
no
Y≤0.0
yes
no
yes
Y>0.5
no
CATG1(I+K)=
CATG1(I+K)+1
yes
Y>1.0
no
CATG2(I+K)=
CATG2(I+K)+1
yes
Y>2.0
no
CATG3(I+K)=
CATG3(I+K)+1
yes
Y>3.0
no
CATG4(I+K)=
CATG4(I+K)+1
yes
Y>4.0
no
CATG5(I+K)=
CATG5(I+K)+1
yes
Y>5.0
no
CATG6(I+K)=
CATG6(I+K)+1
yes
Y>10.0
no
CATG7(I+K)=
CATG7(I+K)+1
CATG8(I+K)=
CATG8(I+K)+1
yes
K=K+2
LOOP
Return
to
calling routin

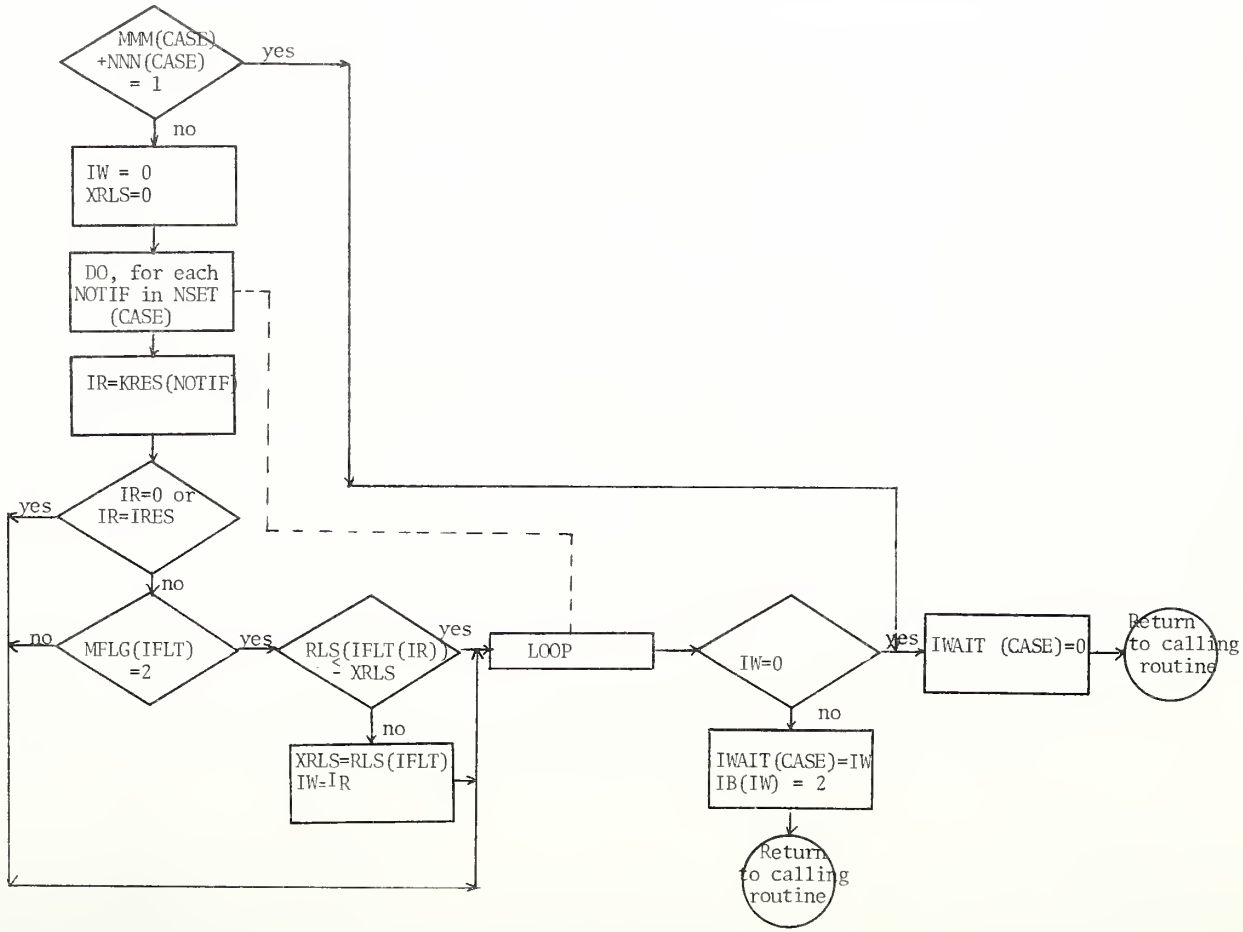
SUBROUTINE SRCHF(CASE, INOTF, IRS)



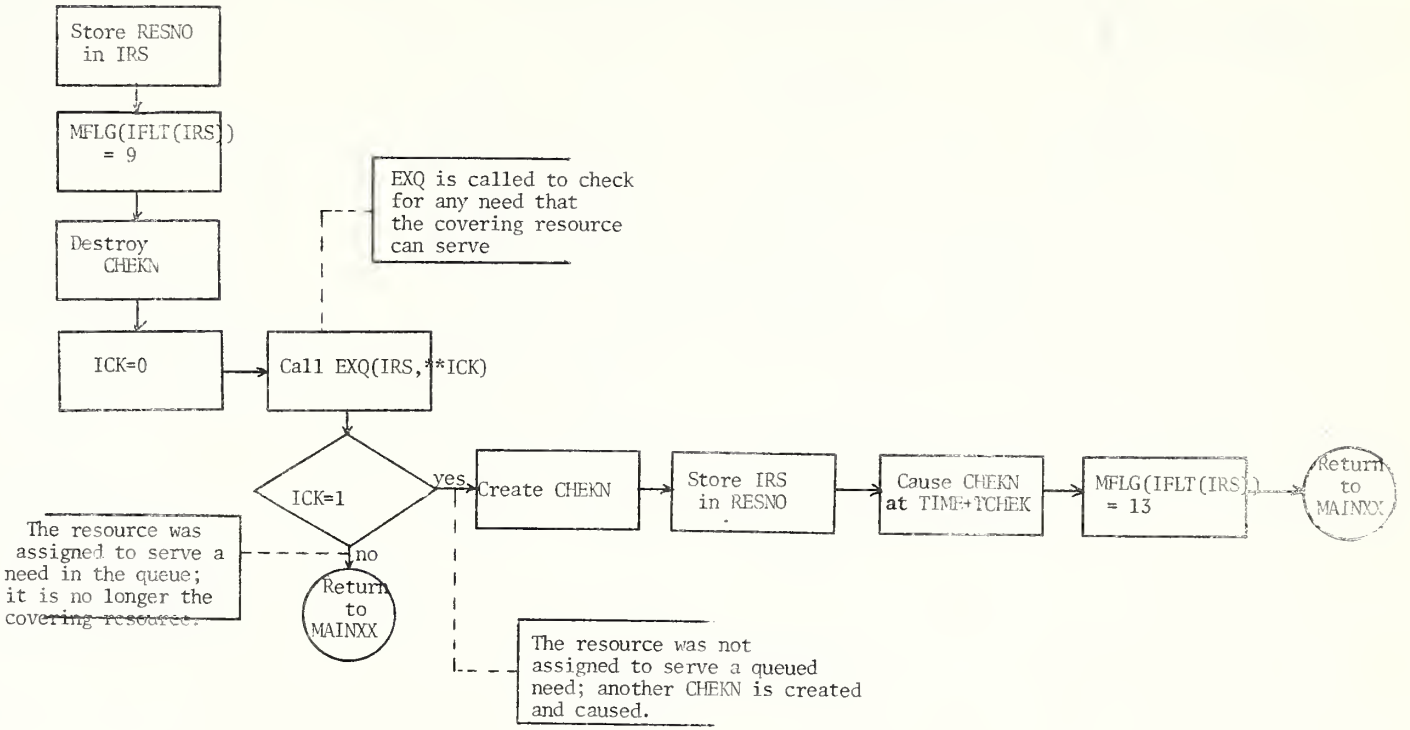
Endogenous Event ONSCN

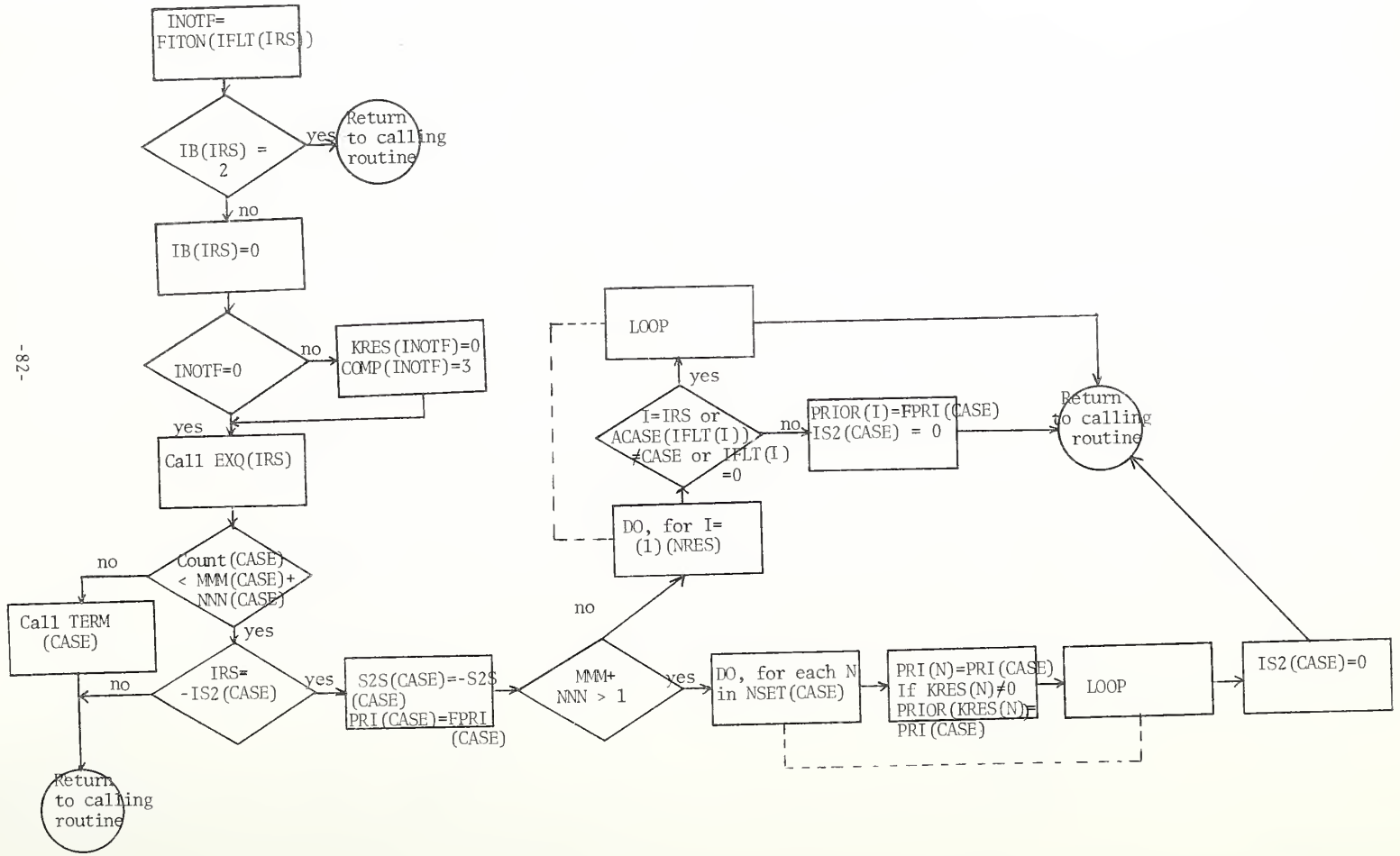


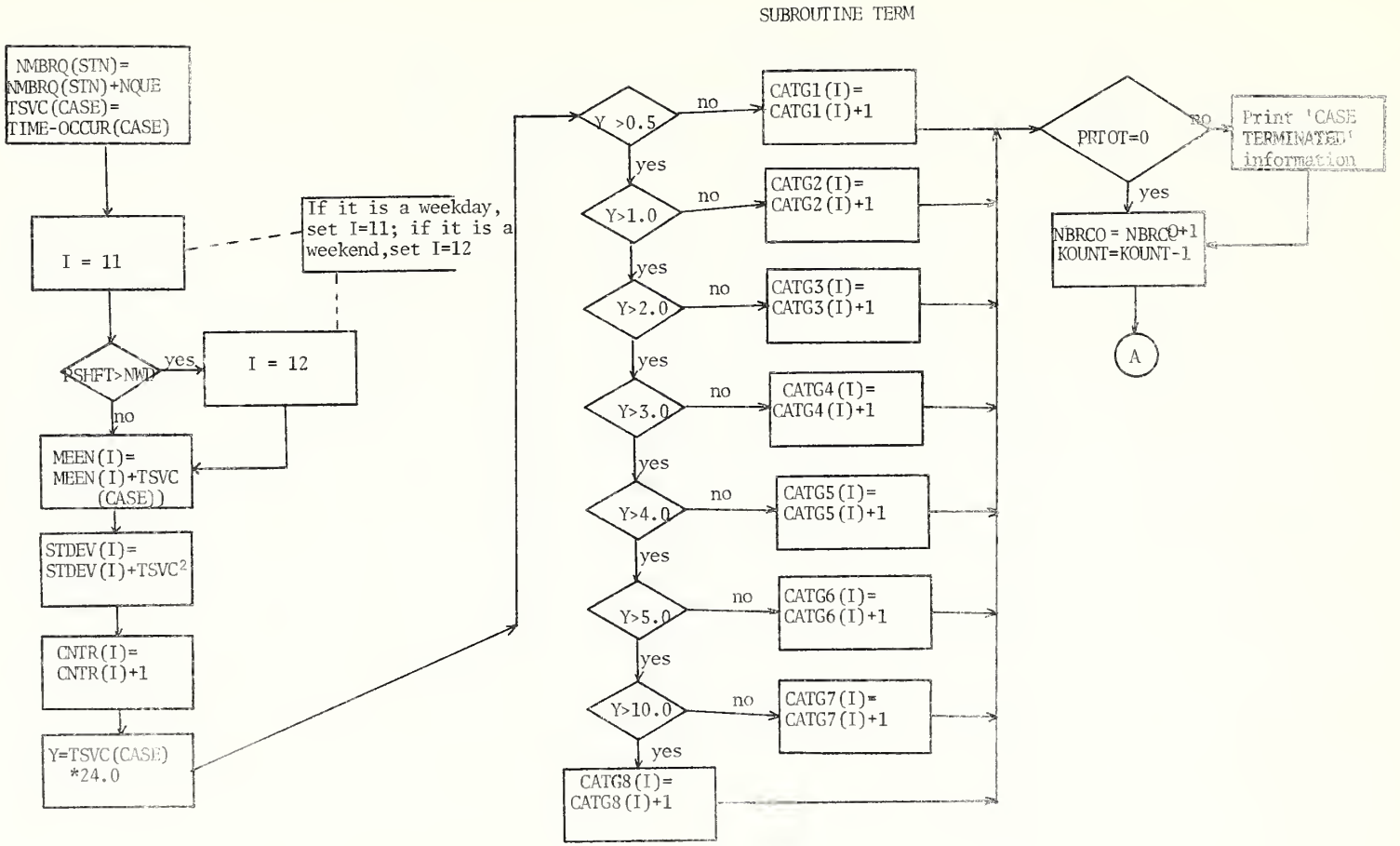

A
IB≠2
yes
CALL RETN (CASE,IRS)
Return to MAINXX
The resource is now covering only
no
CQUE is empty ?
yes
Create CHEKN
Store IRS in RESNO
Cause CHEKN at TIME+TCHEK
MFLG(IFLT(IRS)) =13
Return to MAINXX
B
no
DO, FOR EACH ICNN IN CQUE
No queued need was found for the resource to serve; create and cause a CHEKN for the resource.
Check for any queued need of the same case that the covering resource can serve
yes
FLG≠2
no
yes
CAS(ICNN) ≠CASE
no
CALL CRES (CASE, NEED, KONTR, II)
IT=0 or KONTR=0
no
DO, FOR EACH IRES OF RQUE
yes
ITOL(CASE)=7
IRES=IRS
yes
TVEC(IRS)=0.0 KRES(FITON(IFLT(IRS))=0
Call SVQUE
Return to MAINXX
no
CALL WRECK(ICNN)
LOOP
LOOP
B

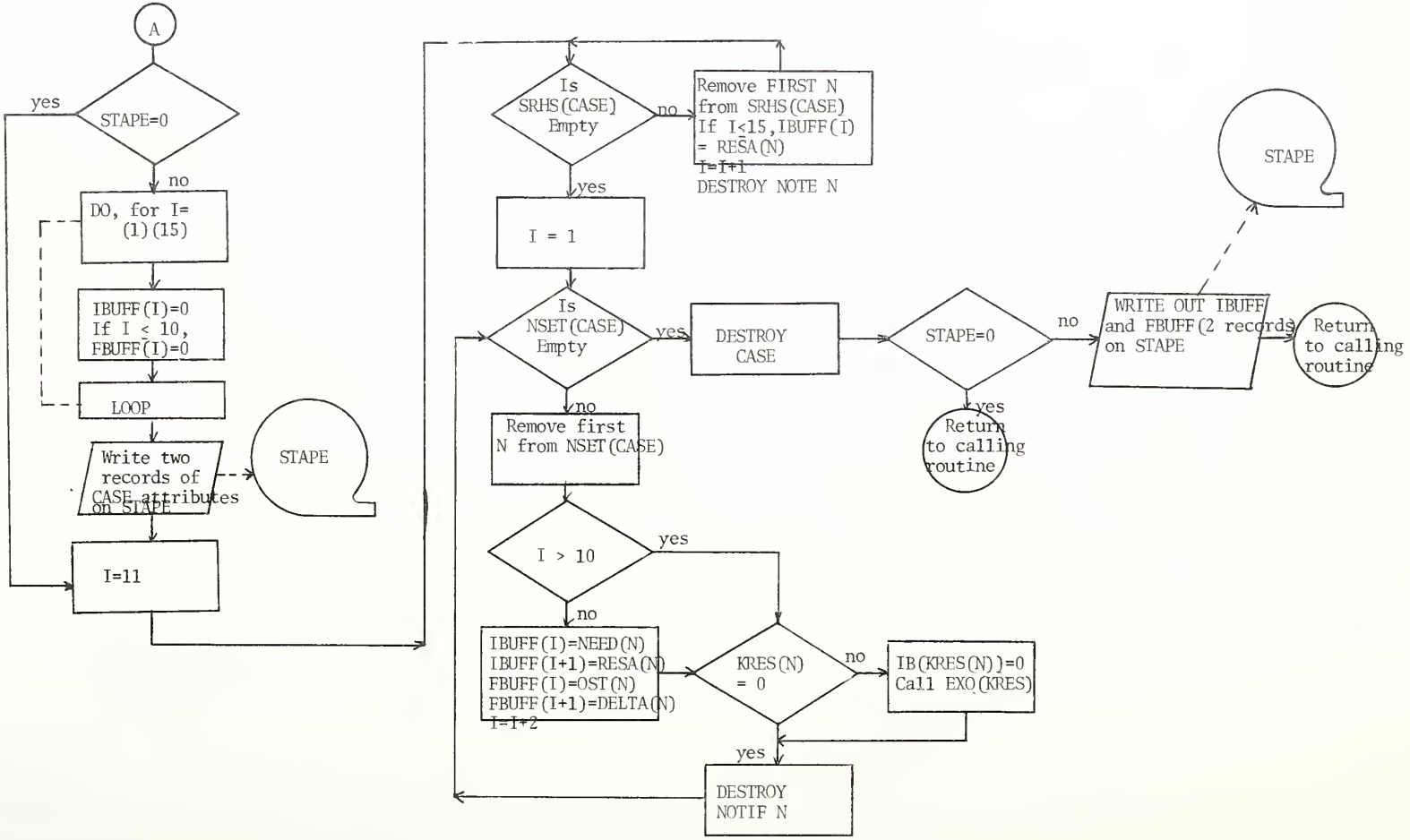

MMM(CASE)
+NNN(CASE)
= 1
yes
no
IW = 0
XRLS=0
DO, for each
NOTIF in NSET
(CASE)
IR=KRES(NOTIF)
IR=0 or
IR=IRES
yes
no
MFLG(IFLT)
=2
no
yes
RLS(IFLT(IR))
< XRLS
yes
no
XRLS=RLS(IFLT)
IW=IR
LOOP
IW=0
yes
no
IWAIT (CASE)=0
Return
to calling
routine
IWAIT(CASE)=IW
IB(IW) = 2
Return
to calling
routine

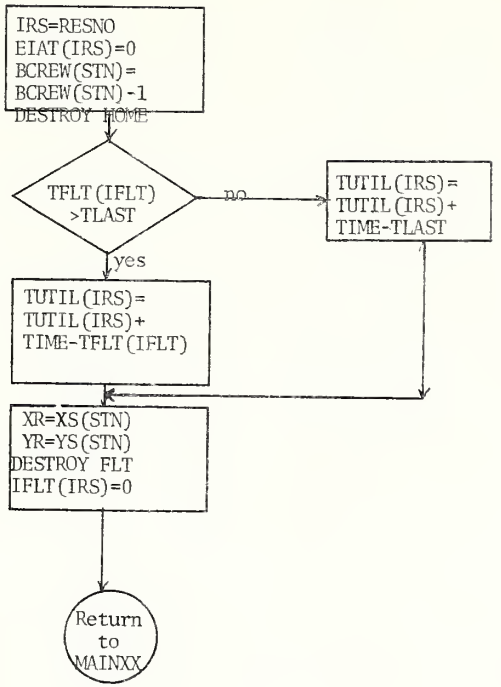
Endogenous Event CHEKN

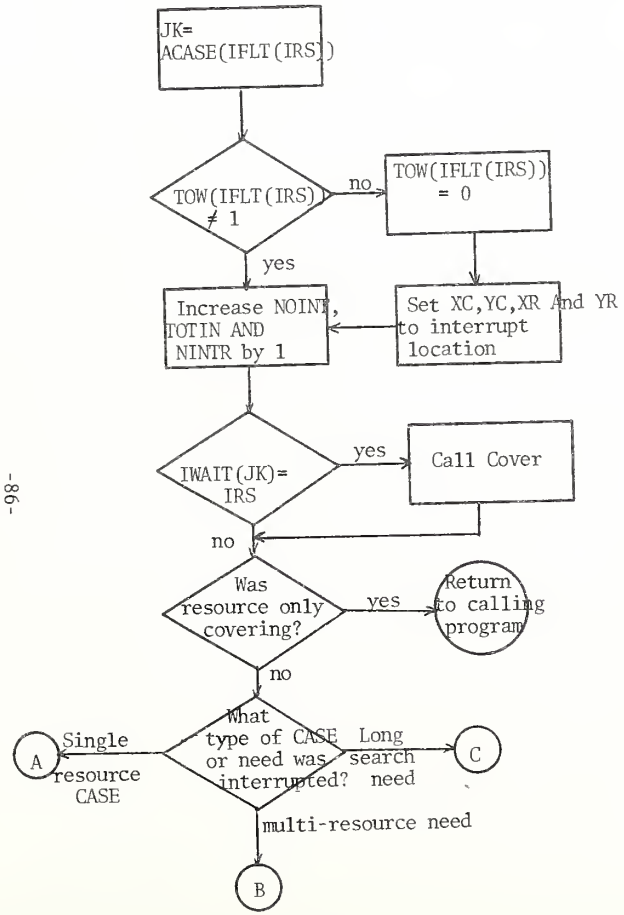

Store RESNO in IRS
MFLG(IFLT(IRS)) = 9
Destroy CHEKN
ICK=0
Call EXQ(IRS,**ICK)
EXQ is called to check for any need that the covering resource can serve
ICK=1
yes
no
Create CHEKN
Store IRS in RESNO
Cause CHEKN at TIME+TCHEK
MFLG(IFLT(IRS)) = 13
Return to MAINXX
The resource was assigned to serve a need in the queue; it is no longer the covering resource.
Return to MAINXX
The resource was not assigned to serve a queued need; another CHEKN is created and caused.


INOTF=
FITON(IFLT(IRS))
IB(IRS) = 2
yes
Return to calling routine
no
IB(IRS)=0
INOTF=0
no
KRES(INOTF)=0
COMP(INOTF)=3
yes
Call EXQ(IRS)
Count(CASE) < MMM(CASE)+ NNN(CASE)
no
Call TERM (CASE)
yes
IRS= -IS2(CASE)
no
yes
S2S(CASE)=-S2S (CASE)
PRI(CASE)=FPRI (CASE)
MMM+ NNN > 1
no
yes
DO, for I= (1)(NRES)
LOOP
I=IRS or ACASE(IFLT(I)) ≠CASE or IFLT(I) =0
yes
no
PRIOR(I)=FPRI(CASE)
IS2(CASE) = 0
Return to calling routine
DO, for each N in NSET(CASE)
PRI(N)=PRI(CASE)
If KRES(N)≠0
PRIOR(KRES(N))= PRI(CASE)
LOOP
IS2(CASE)=0
Return to calling routine

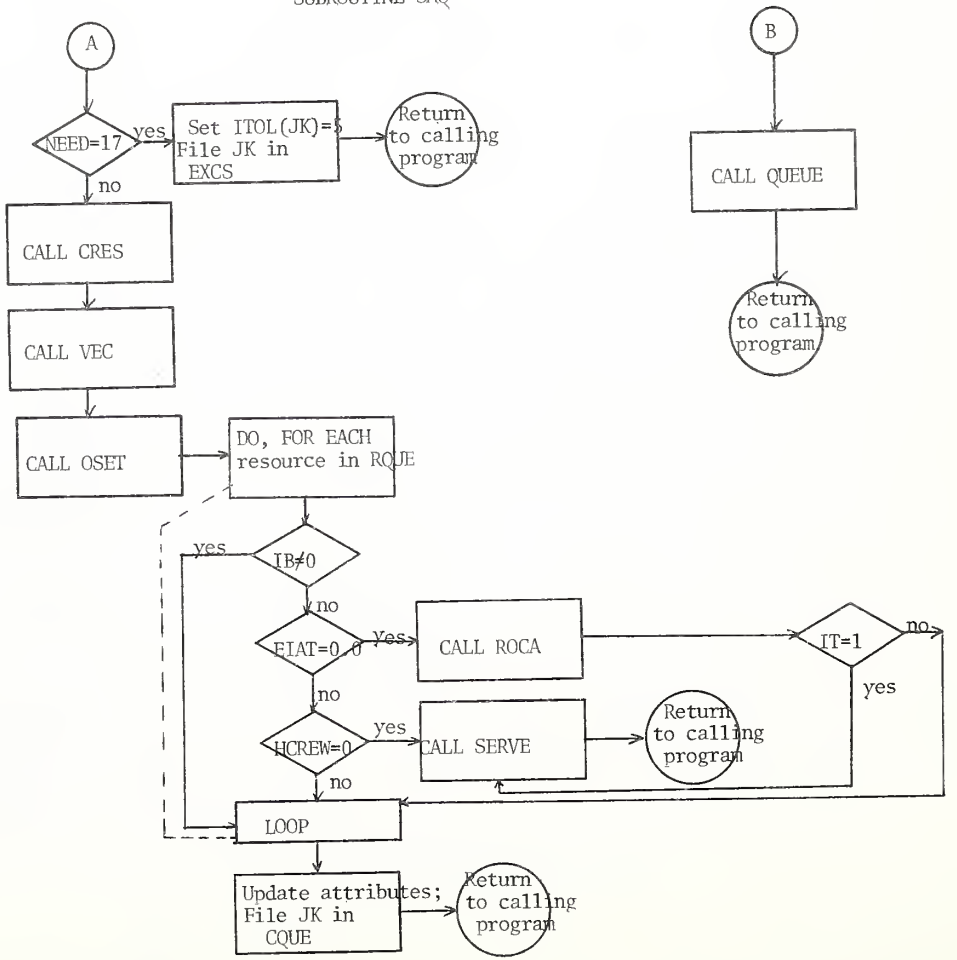

SUBROUTINE TERM
NMBRQ(STN)=
NMBRQ(STN)+NQUE
TSVC(CASE)=
TIME-OCCUR(CASE)
I = 11
If it is a weekday, set I=11; if it is a weekend,set I=12
PSHFT>NWD
yes
I = 12
no
MEEN(I)=
MEEN(I)+TSVC (CASE))
STDEV(I)=
STDEV(I)+TSVC2
CNTR(I)=
CNTR(I)+1
Y=TSVC(CASE) *24.0
Y >0.5
no
CATG1(I)=
CATG1(I)+1
yes
Y>1.0
no
CATG2(I)=
CATG2(I)+1
yes
Y>2.0
no
CATG3(I)=
CATG3(I)+1
yes
Y>3.0
no
CATG4(I)=
CATG4(I)+1
yes
Y>4.0
no
CATG5(I)=
CATG5(I)+1
yes
Y>5.0
no
CATG6(I)=
CATG6(I)+1
yes
Y>10.0
no
CATG7(I)=
CATG7(I)+1
yes
CATG8(I)=
CATG8(I)+1
PRTOT=0
no
Print 'CASE TERMINATED' information
yes
NBRCO = NBRCO+1
KOUNT=KOUNT-1
A

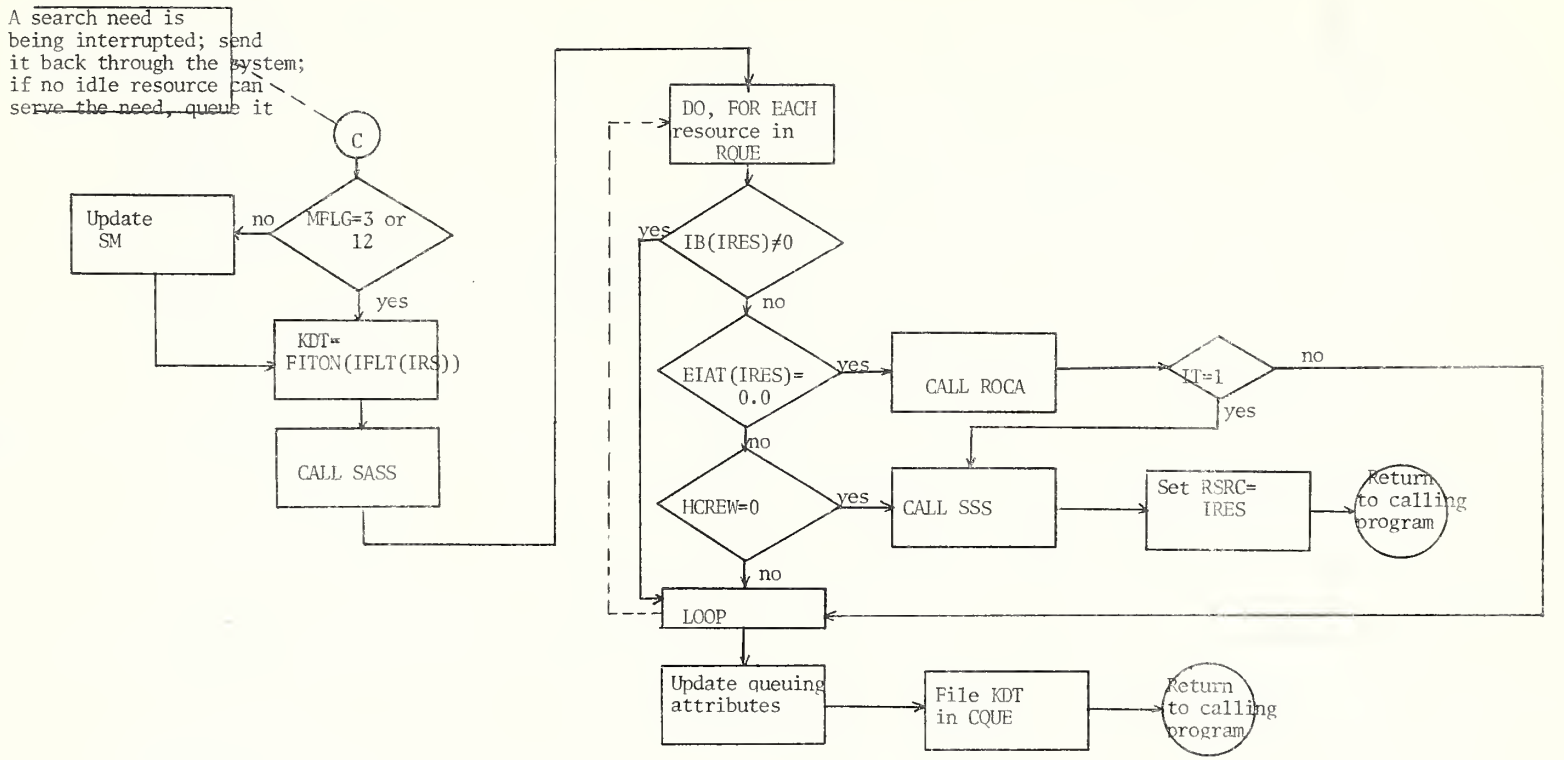

A
yes
STAPE=0
no
DO, for I=
(1)(15)
IBUFF(I)=0
If I < 10,
FBUFF(I)=0
LOOP
Write two
records of
CASE attributes
on STAPE
STAPE
I=11
Is
SRHS(CASE)
Empty
no
Remove FIRST N
from SRHS(CASE)
If I≤15, IBUFF(I)
= RESA(N)
I=I+1
DESTROY NOTE N
yes
I = 1
Is
NSET(CASE)
Empty
yes
DESTROY
CASE
STAPE=0
no
WRITE OUT IBUFF
and FBUFF(2 records)
on STAPE
STAPE
Return
to calling
routine
yes
Return
to calling
routine
no
Remove first
N from NSET(CASE)
I > 10
yes
no
IBUFF(I)=NEED(N)
IBUFF(I+1)=RESA(N)
FBUFF(I)=OST(N)
FBUFF(I+1)=DELTA(N)
I=I+2
KRES(N)
= 0
no
IB(KRES(N))=0
Call EXQ(KRES)
yes
DESTROY
NOTIF N

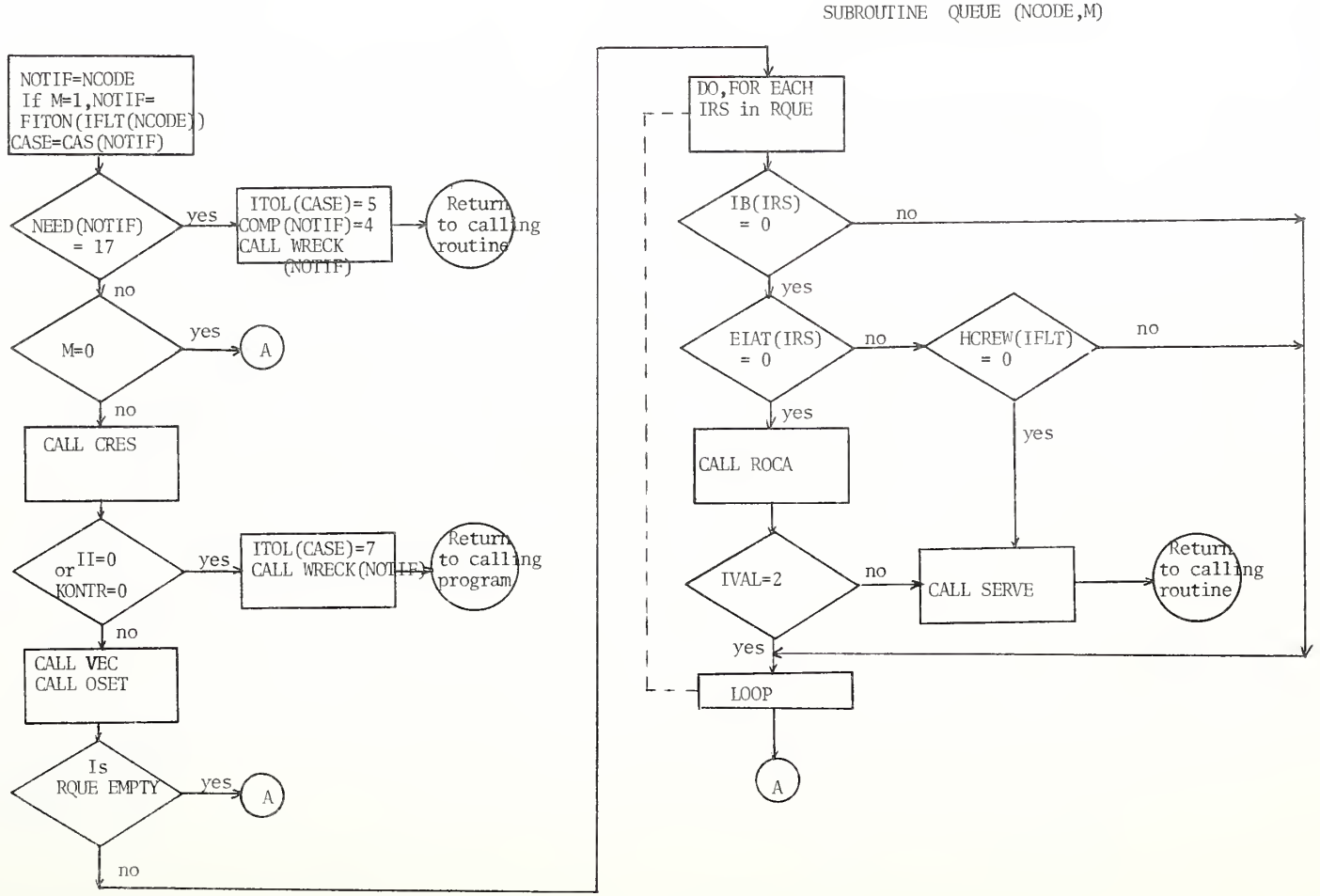

IRS=RESNO
EIAT(IRS)=0
BCREW(STN)=
BCREW(STN)-1
DESTROY HOME
TFLT(IFLT)
>TLAST
no
TUTIL(IRS)=
TUTIL(IRS)+
TIME-TLAST
yes
TUTIL(IRS)=
TUTIL(IRS)+
TIME-TFLT(IFLT)
XR=XS(STN)
YR=YS(STN)
DESTROY FLT
IFLT(IRS)=0
Return
to
MAINXX



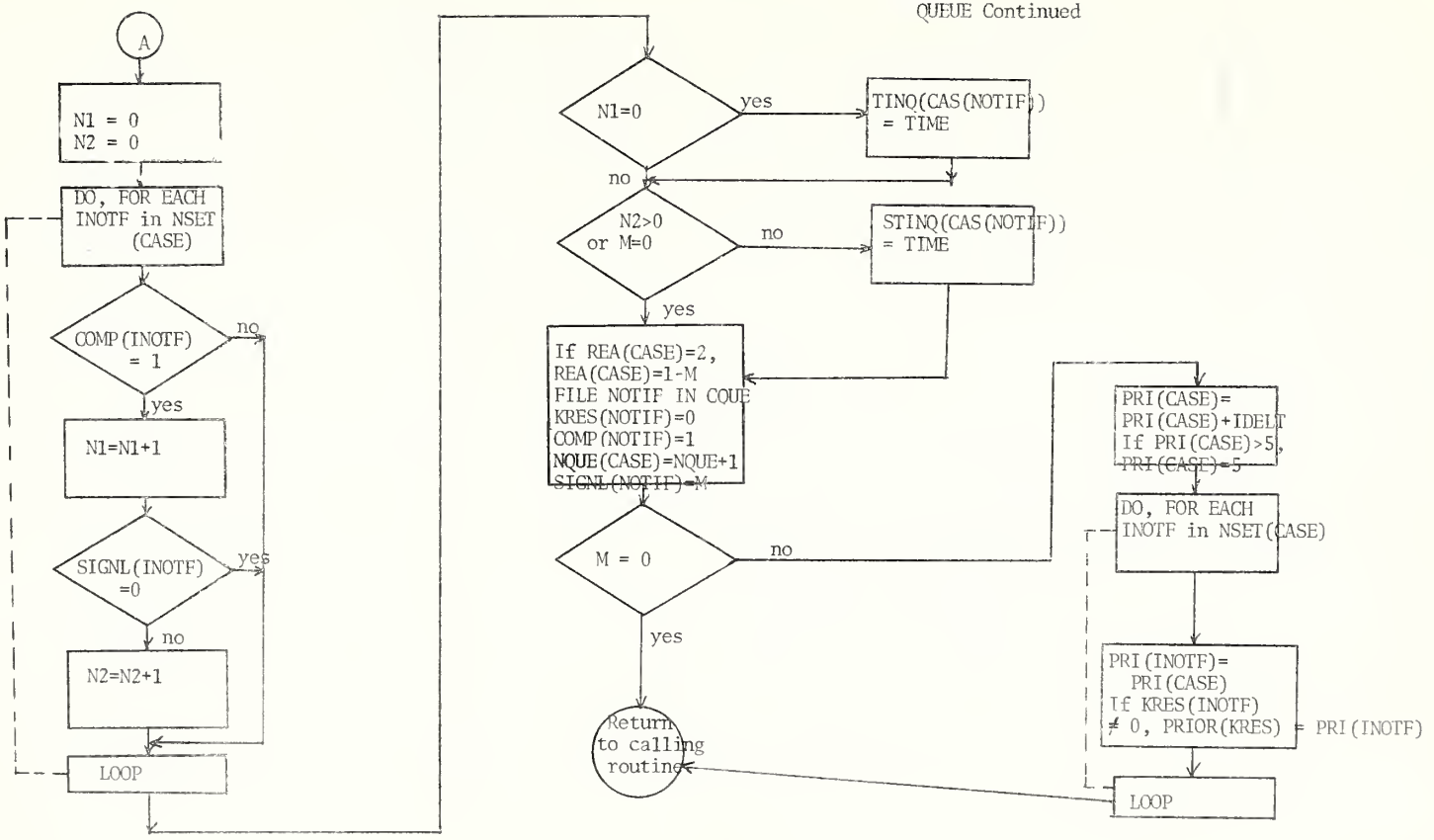
SUBROUTINE SAQ



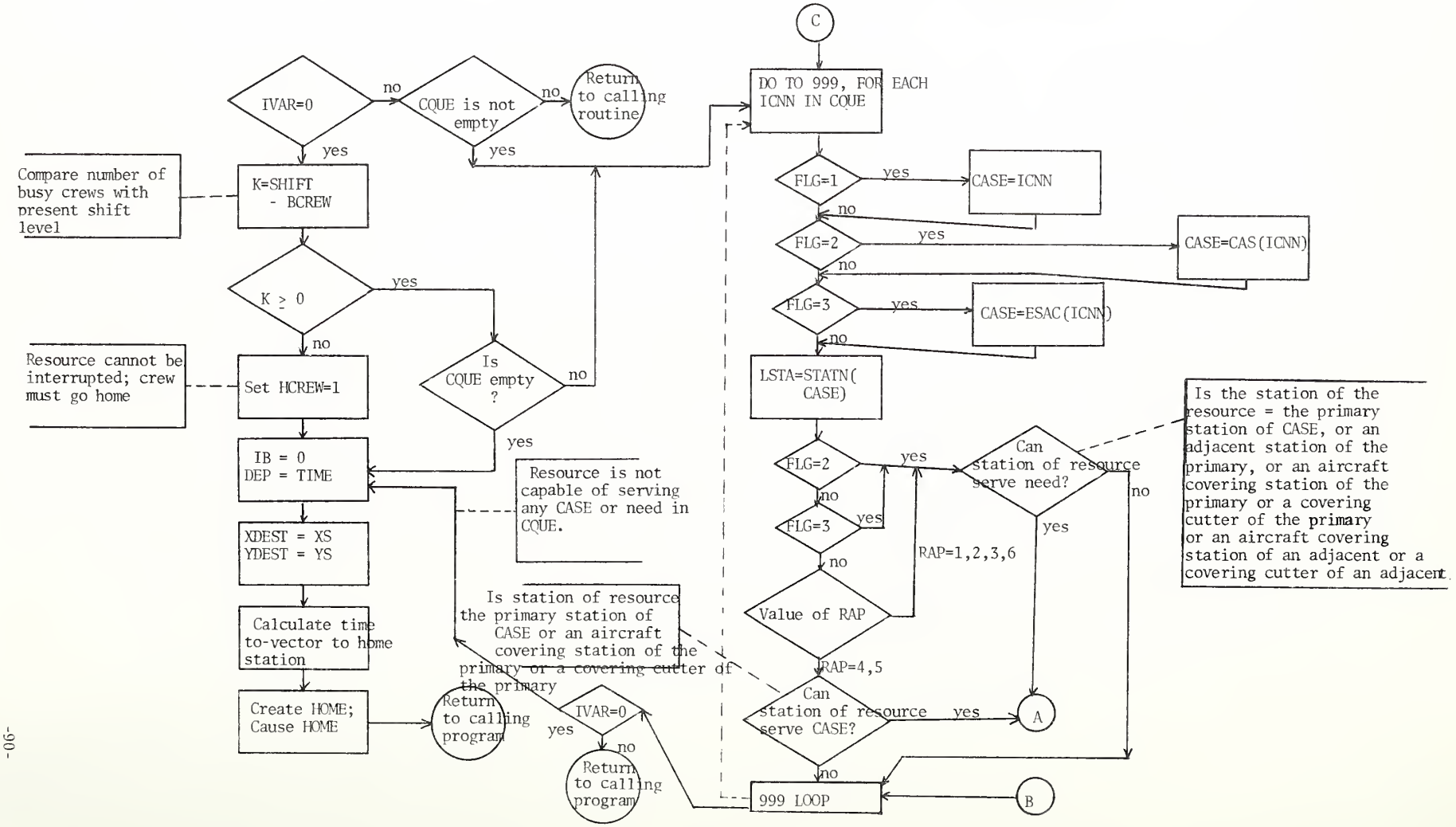

A search need is being interrupted; send it back through the system; if no idle resource can serve the need, queue it
C
MFLG=3 or 12
no
Update SM
yes
KDT= FITON(IFLT(IRS))
CALL SASS
DO, FOR EACH resource in RQUE
IB(IRES)≠0
yes
no
EIAT(IRES)= 0.0
yes
CALL ROCA
IT=1
no
yes
no
HCREW=0
yes
CALL SSS
Set RSRC= IRES
Return to calling program
no
LOOP
Update queuing attributes
File KDT in CQUE
Return to calling program


NOTIF=NCODE
If M=1,NOTIF=
FITON(IFLT(NCODE))
CASE=CAS(NOTIF)
NEED(NOTIF) = 17
yes
ITOL(CASE)=5
COMP(NOTIF)=4
CALL WRECK (NOTIF)
Return to calling routine
no
M=0
yes
A
no
CALL CRES
II=0 or KONTR=0
yes
ITOL(CASE)=7
CALL WRECK(NOTIF)
Return to calling program
no
CALL VEC
CALL OSET
Is RQUE EMPTY
yes
A
no
SUBROUTINE QUEUE (NCODE,M)
DO,FOR EACH IRS in RQUE
IB(IRS) = 0
no
yes
EIAT(IRS) = 0
no
HCREW(IFLT) = 0
no
yes
yes
CALL ROCA
IVAL=2
no
CALL SERVE
Return to calling routine
yes
LOOP
A

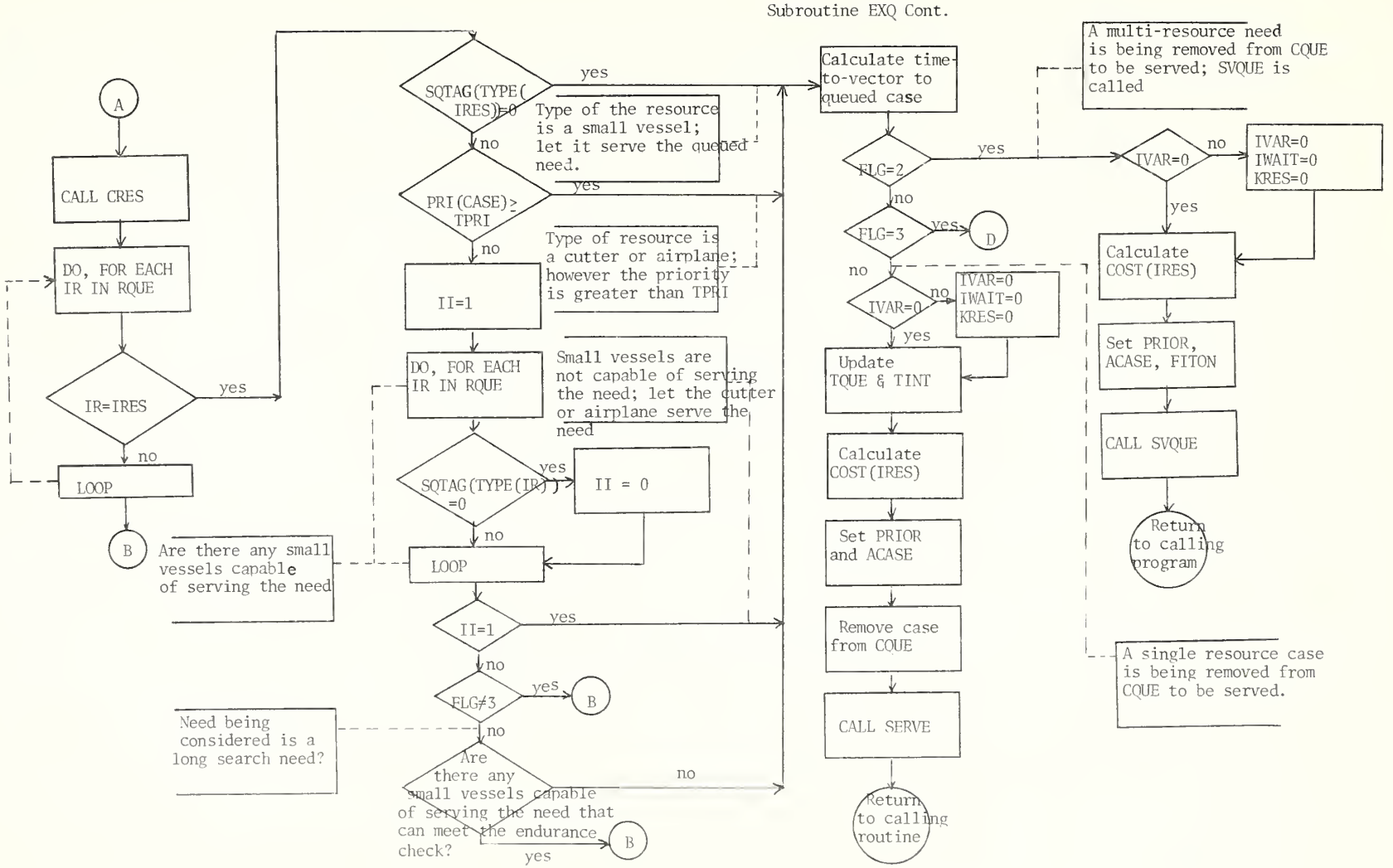
QUEUE Continued

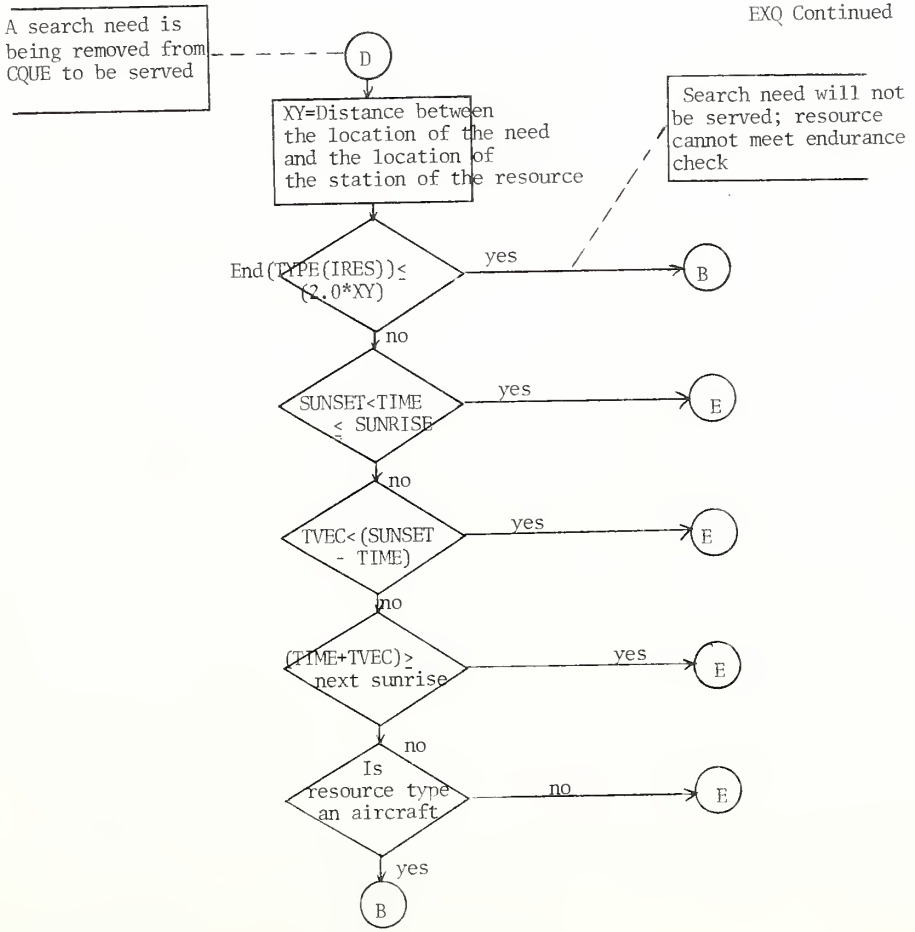


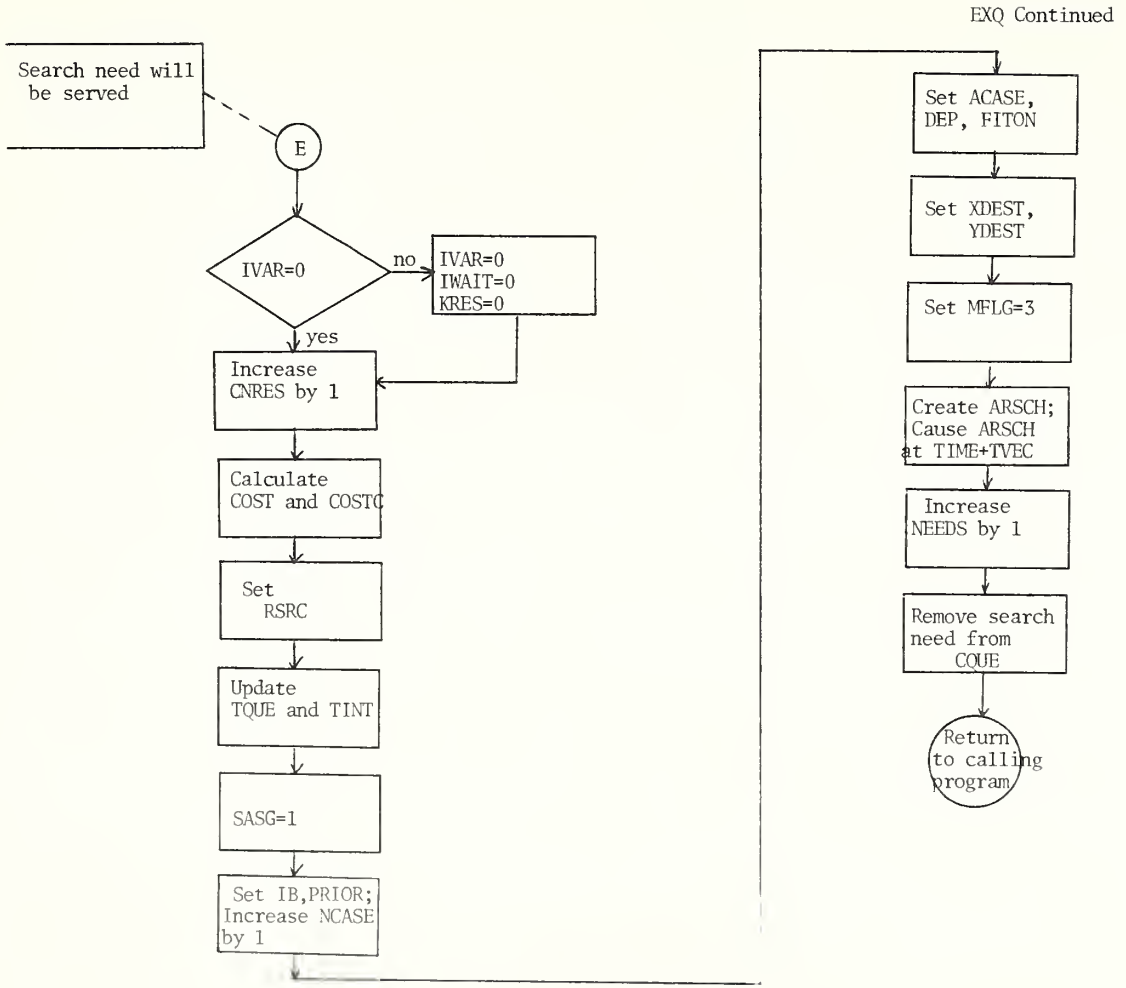
SUBROUTINE EXQ



Subroutine EXQ Cont.



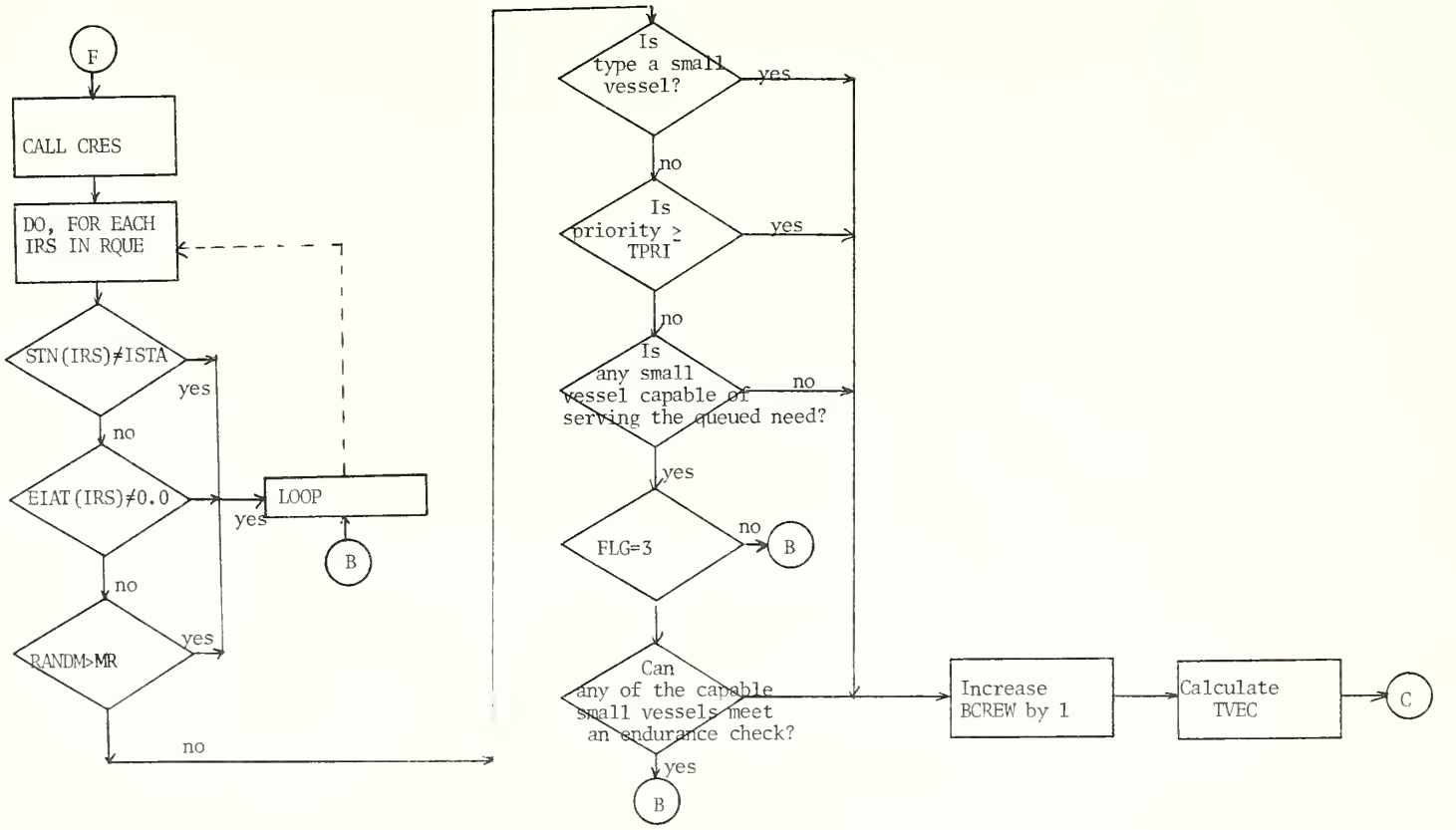

A search need is being removed from CQUE to be served
D
XY=Distance between the location of the need and the location of the station of the resource
Search need will not be served; resource cannot meet endurance check
End(TYPE(IRES))≤ (2.0*XY)
yes
B
no
SUNSET<TIME ≤ SUNRISE
yes
E
no
TVEC<(SUNSET - TIME)
yes
E
no
(TIME+TVEC)≥ next sunrise
yes
E
no
Is resource type an aircraft
no
E
yes
B

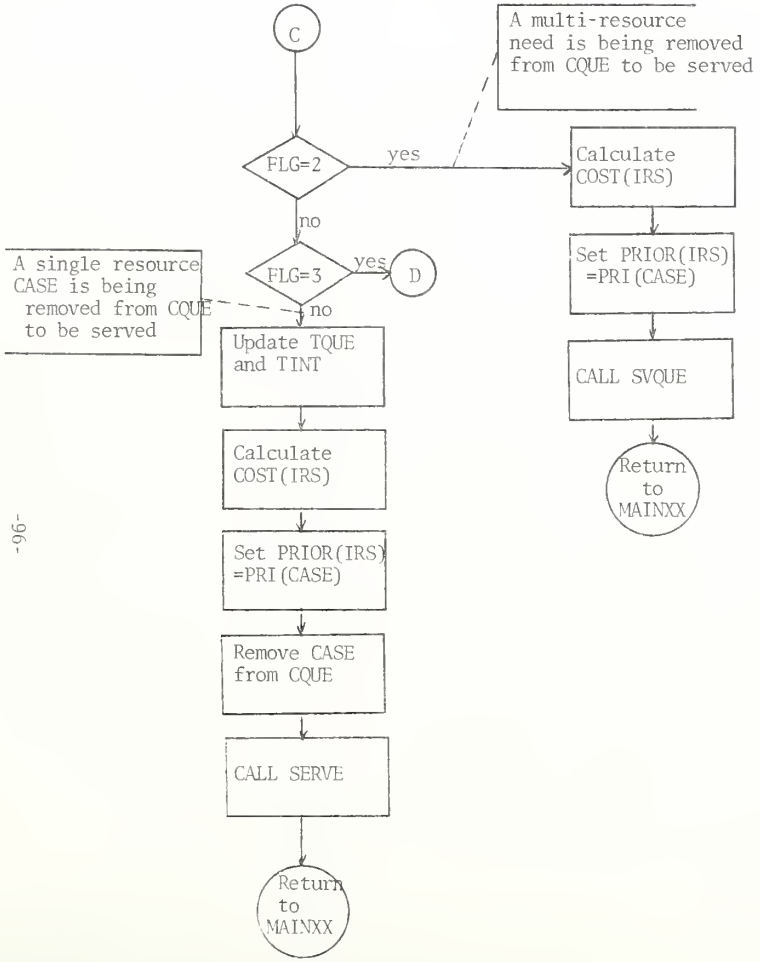


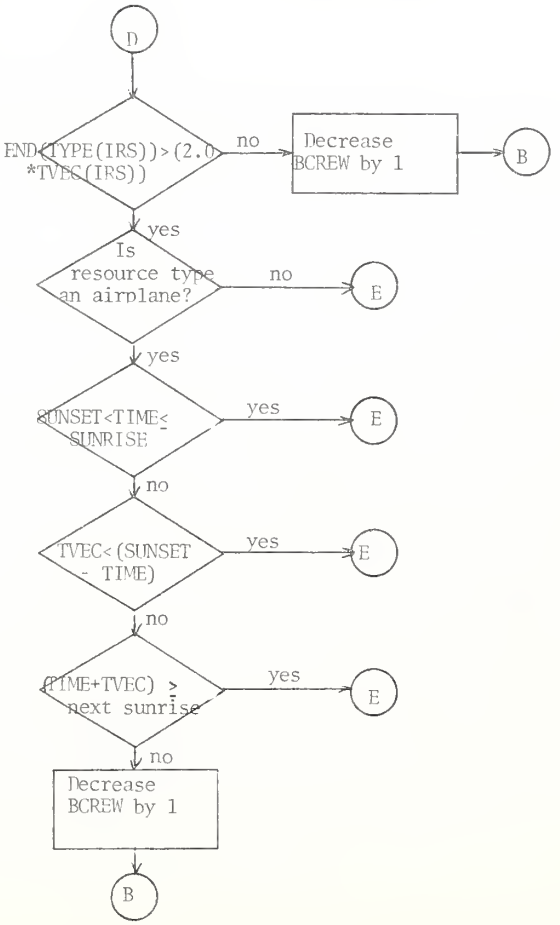
EXQ Continued

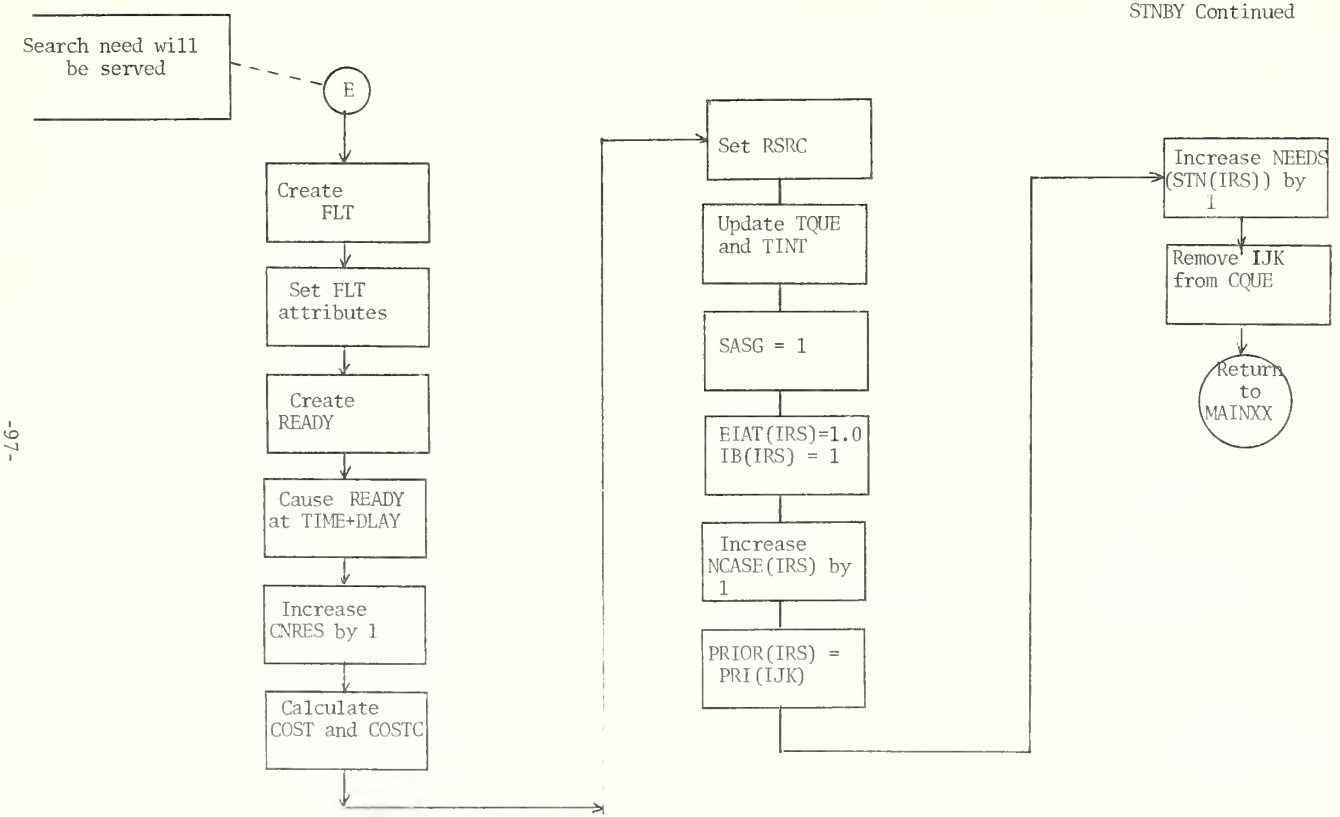
Set ACASE, DEP, FITON
Set XDEST, YDEST
Set MFLG=3
Create ARSCH; Cause ARSCH at TIME+TVEC
Increase NEEDS by 1
Remove search need from CQUE
Return to calling program

Endogenous Event STNBY

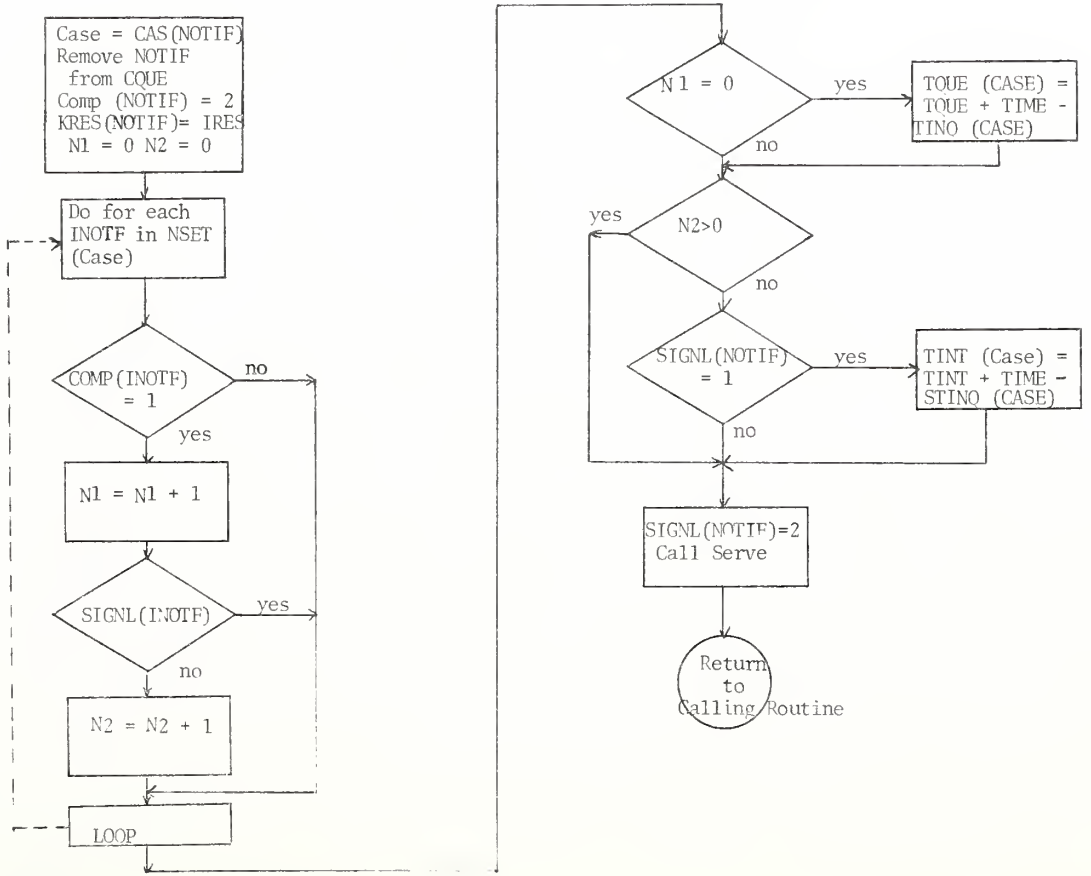

F
CALL CRES
DO, FOR EACH IRS IN RQUE
STN(IRS)≠ISTA
yes
no
EIAT(IRS)≠0.0
yes
LOOP
B
no
RANDM>MR
yes
no
Is type a small vessel?
yes
no
Is priority ≥ TPRI
yes
no
Is any small vessel capable of serving the queued need?
no
yes
FLG=3
no
B
Can any of the capable small vessels meet an endurance check?
yes
B
Increase BCREW by 1
Calculate TVEC
C


C
A multi-resource need is being removed from CQUE to be served
FLG=2
yes
Calculate COST(IRS)
no
Set PRIOR(IRS) =PRI(CASE)
FLG=3
yes
D
A single resource CASE is being removed from CQUE to be served
no
CALL SVQUE
Update TQUE and TINT
Return to MAINXX
Calculate COST(IRS)
Set PRIOR(IRS) =PRI(CASE)
Remove CASE from CQUE
CALL SERVE
Return to MAINXX

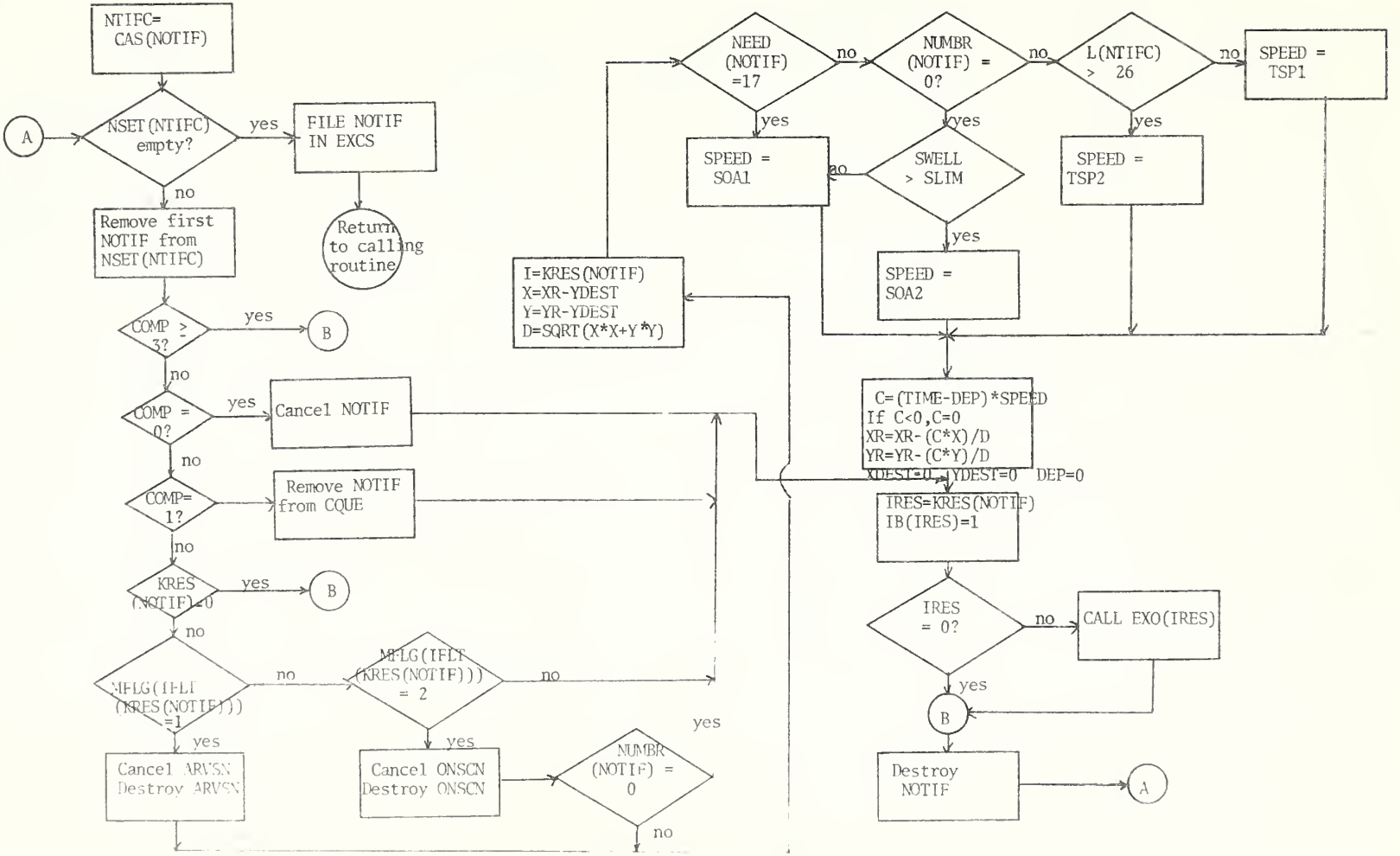

D
END(TYPE(IRS))>(2.0 *TVEC(IRS))
no
Decrease BCREW by 1
B
yes
Is resource type an airplane?
no
E
yes
SUNSET<TIME≤ SUNRISE
yes
E
no
TVEC<(SUNSET - TIME)
yes
E
no
(TIME+TVEC) ≥ next sunrise
yes
E
no
Decrease BCREW by 1
B

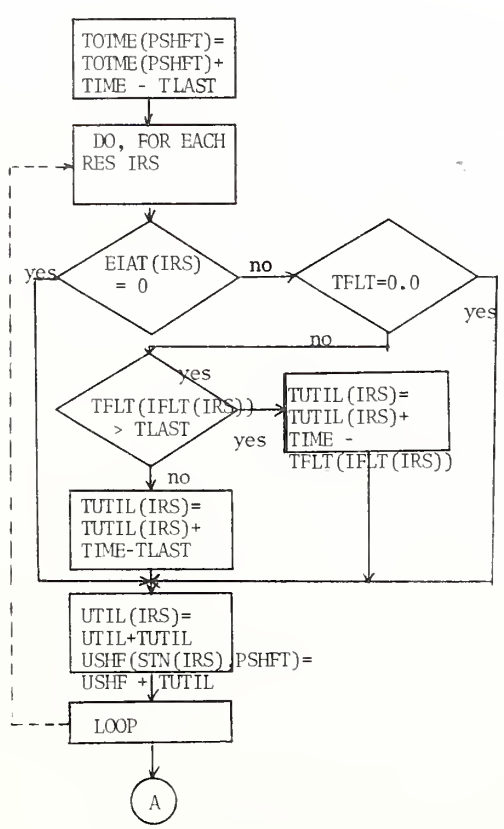

Search need will be served
E
Create FLT
Set FLT attributes
Create READY
Cause READY at TIME+DLAY
Increase CNRES by 1
Calculate COST and COSTC
Set RSRC
Update TQUE and TINT
SASG = 1
EIAT(IRS)=1.0
IB(IRS) = 1
Increase NCASE(IRS) by 1
PRIOR(IRS) = PRI(IJK)
Increase NEEDS (STN(IRS)) by 1
Remove IJK from CQUE
Return to MAINXX

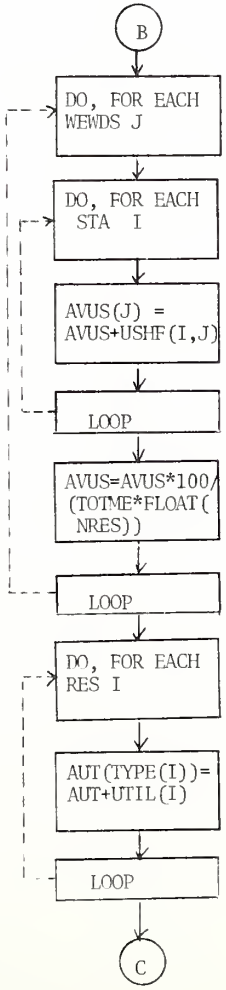
SUBROUTINE SVQUE



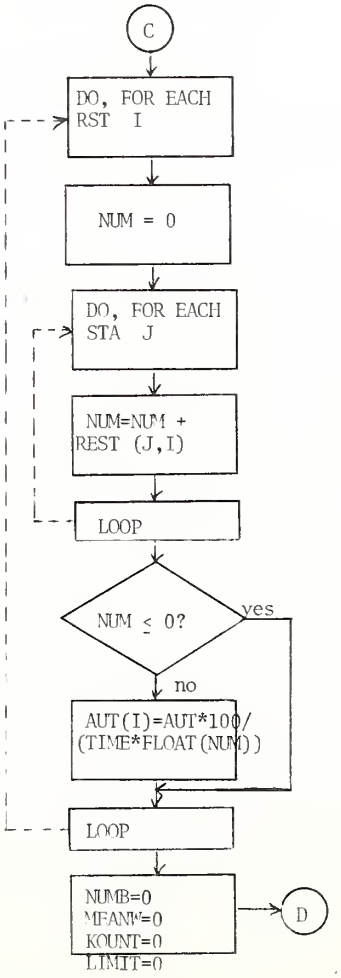
Subroutine WRECK (NOTIF)



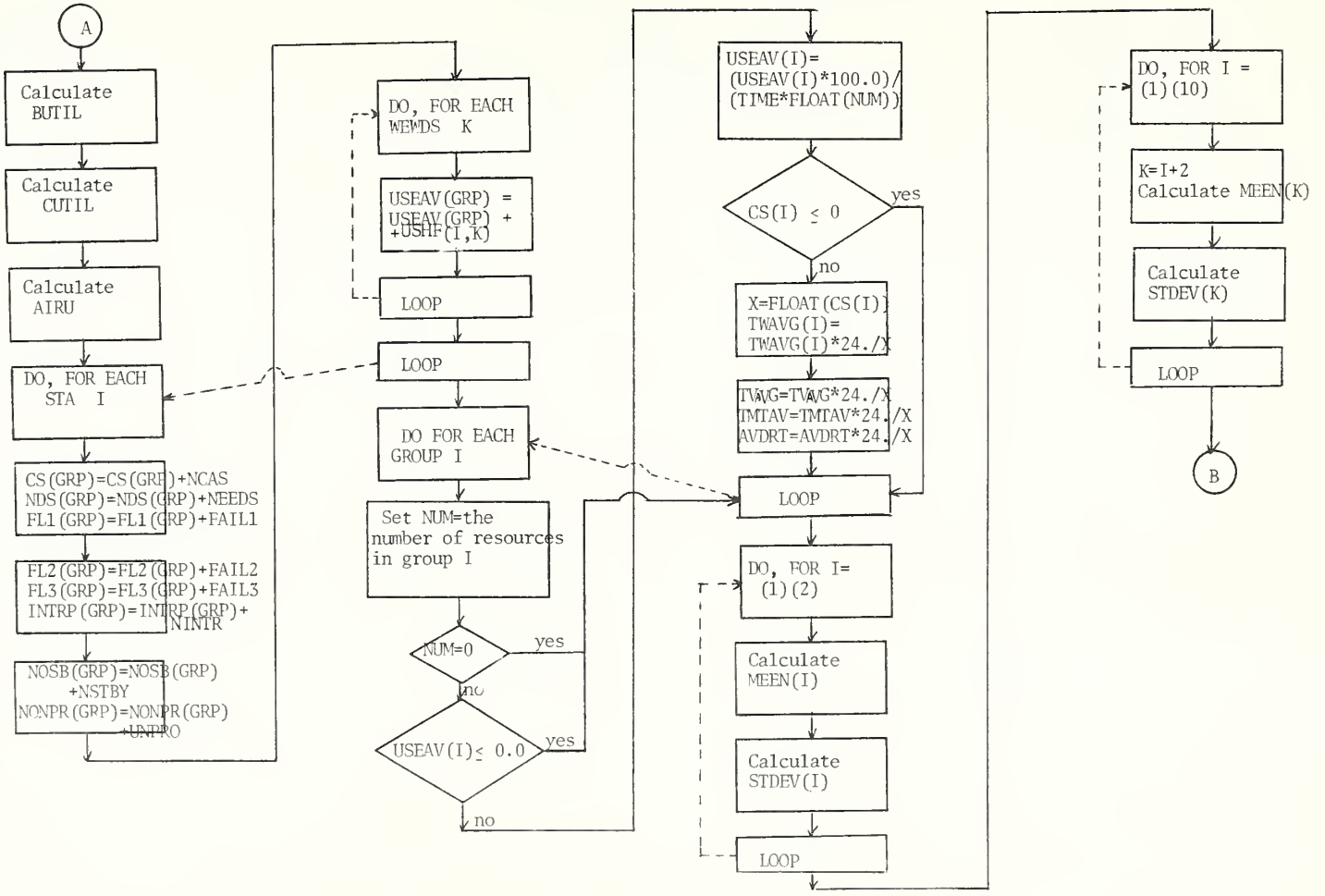




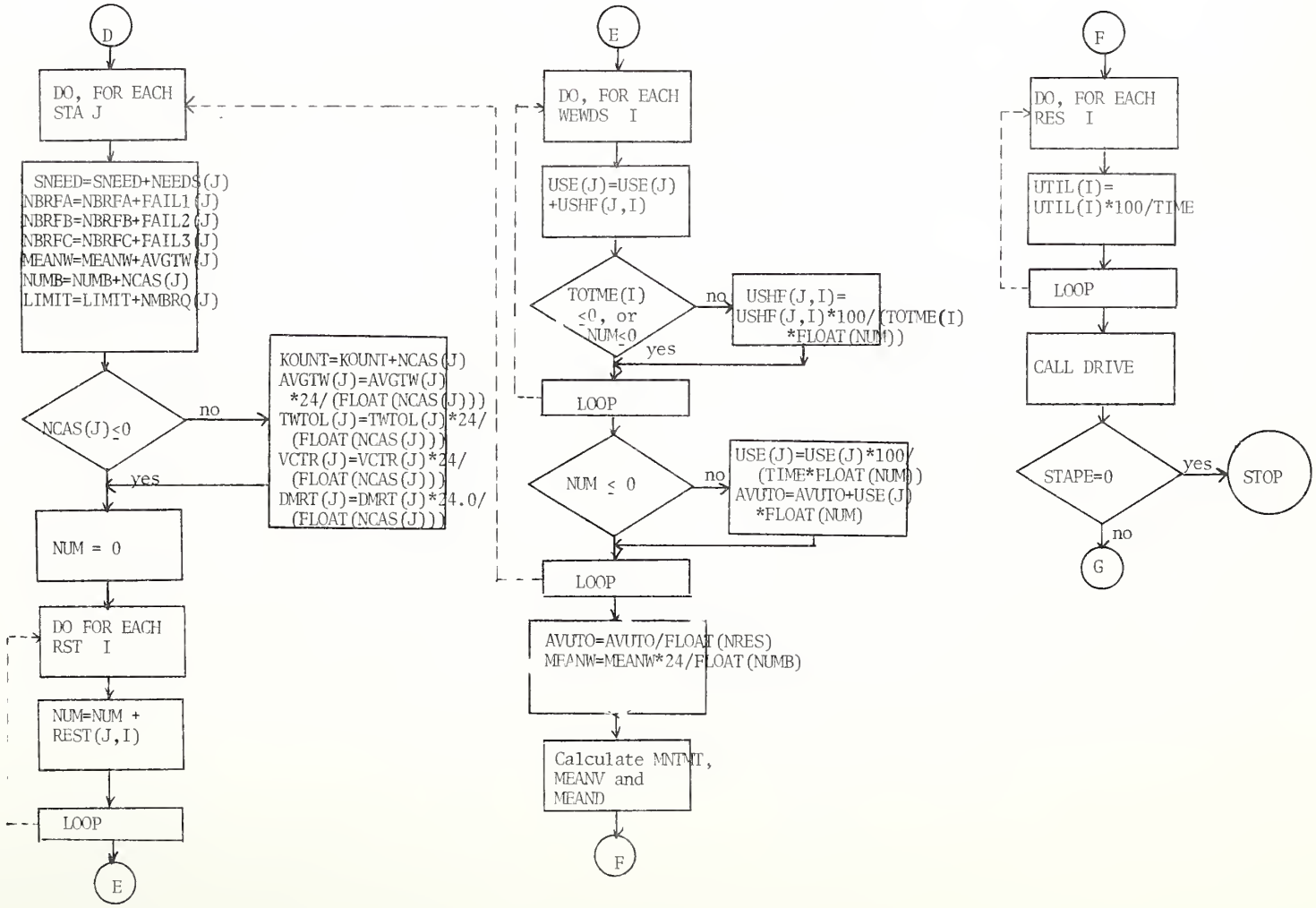
EXOGENOUS EVENT ENDSIM

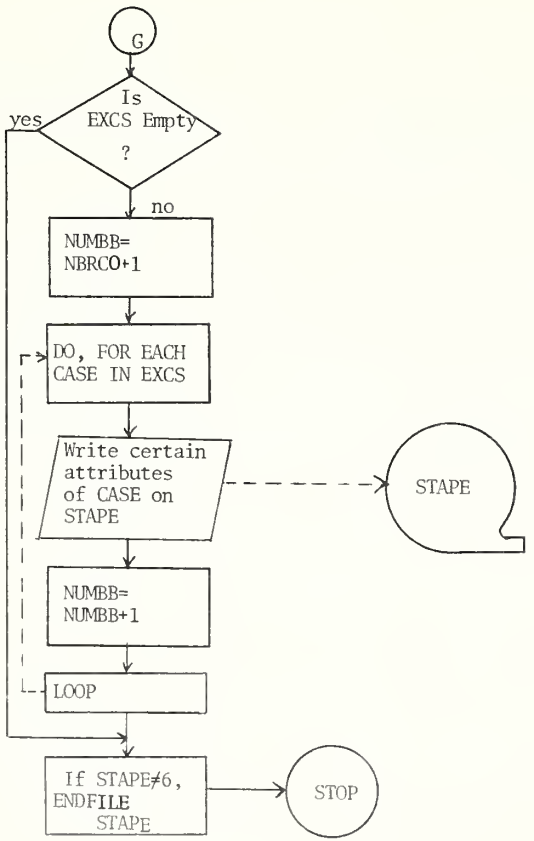


ENDSIM Continued

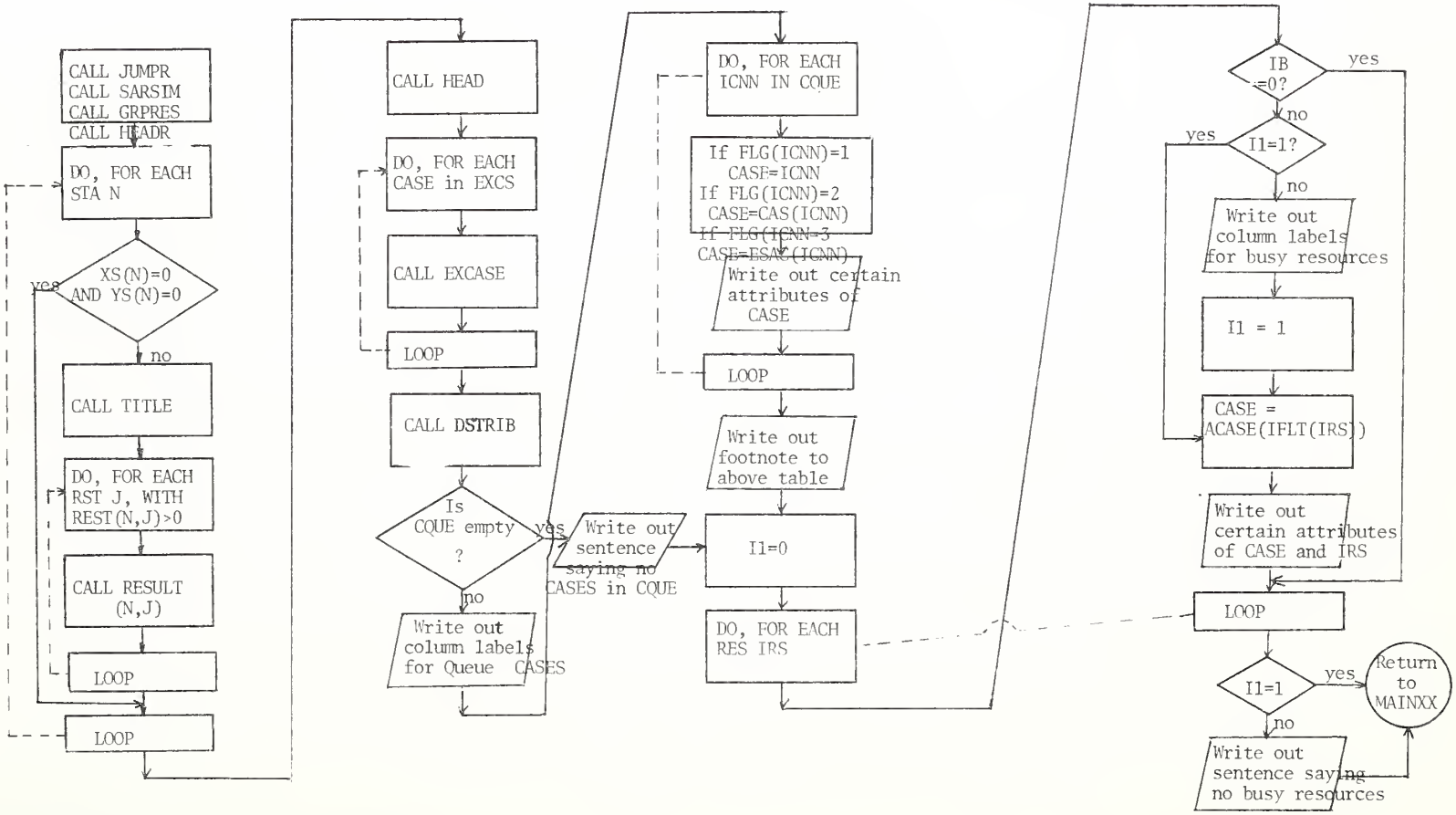


ENDSIM Continued

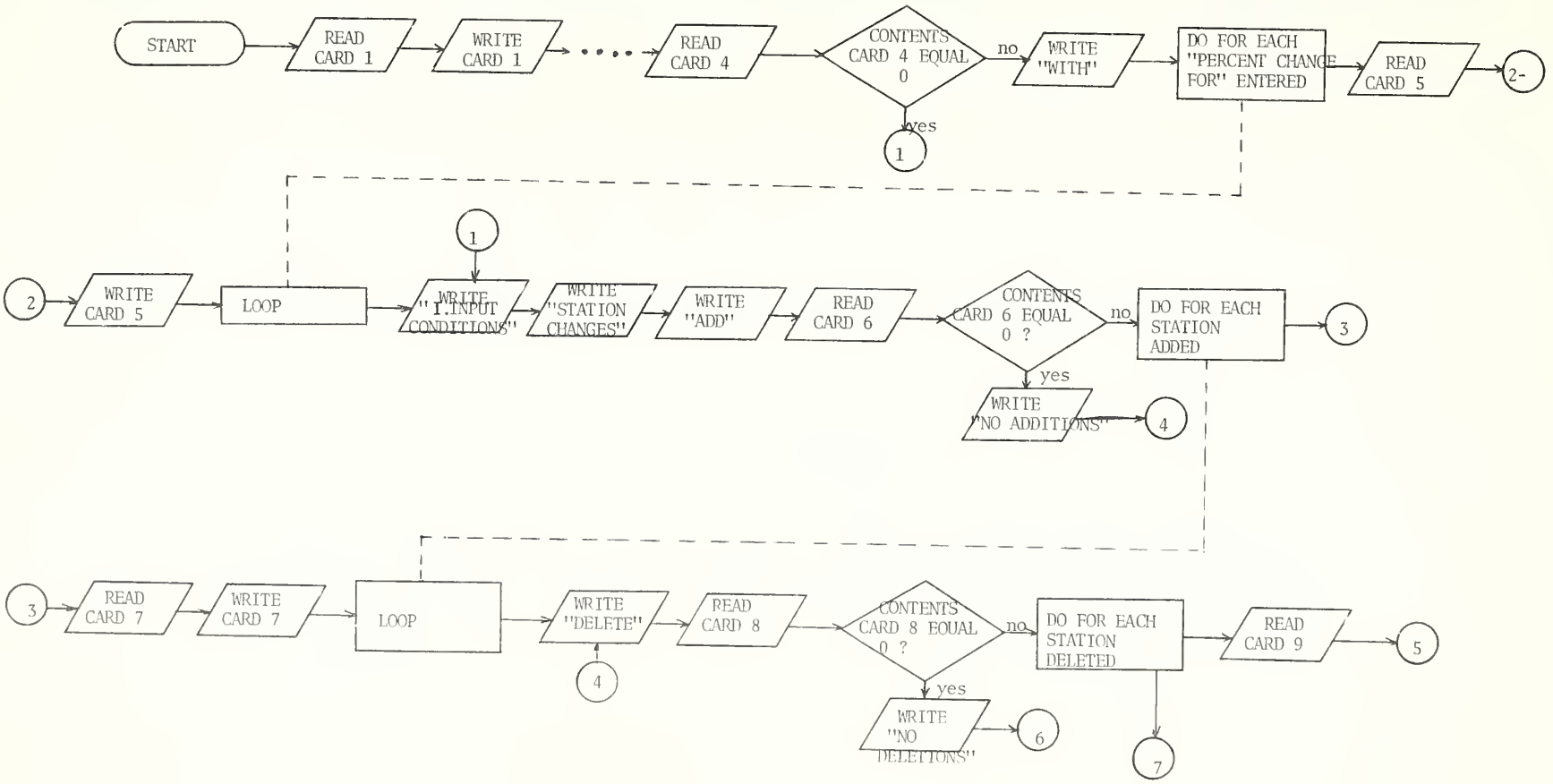



G
Is EXCS Empty ?
yes
no
NUMBB= NBRCO+1
DO, FOR EACH CASE IN EXCS
Write certain attributes of CASE on STAPE
STAPE
NUMBB= NUMBB+1
LOOP
If STAPE≠6, ENDFILE STAPE
STOP

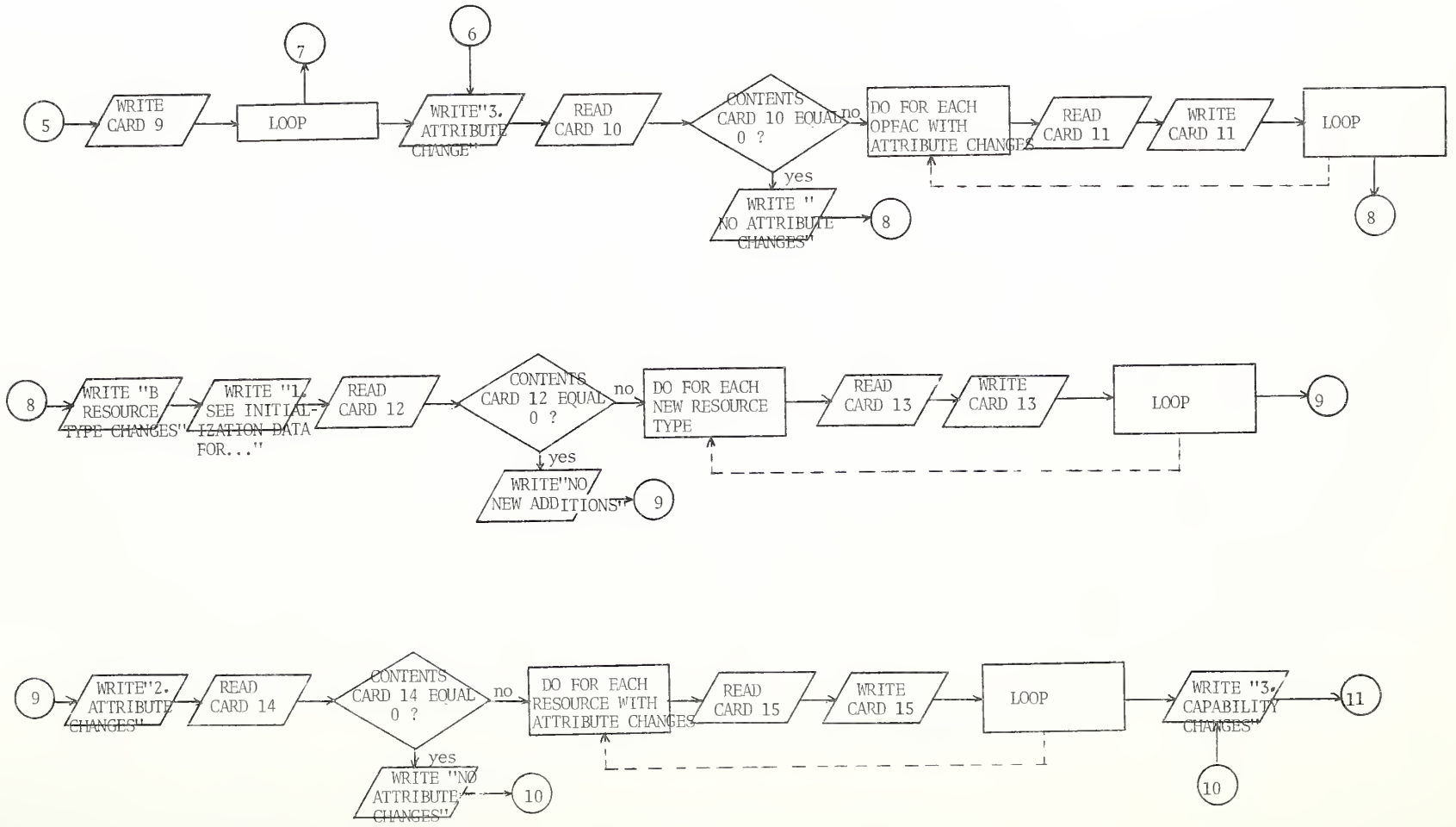
SUBROUTINE DRIVE

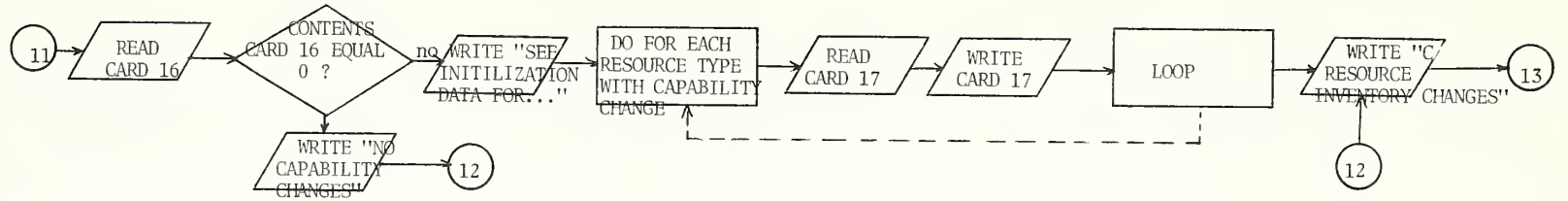


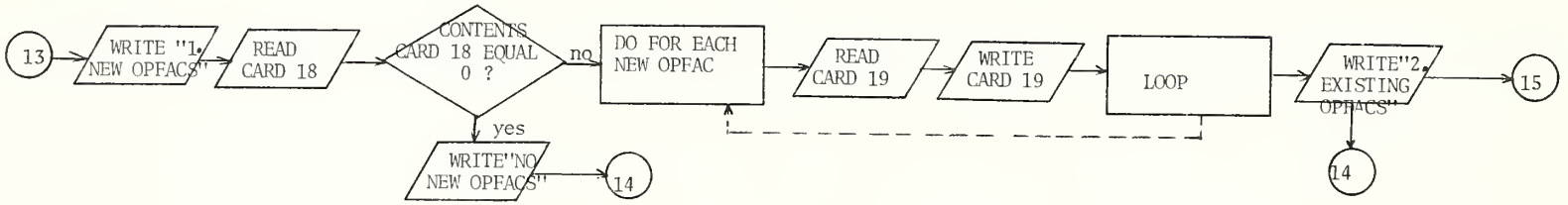
SUBROUTINE JUMPER

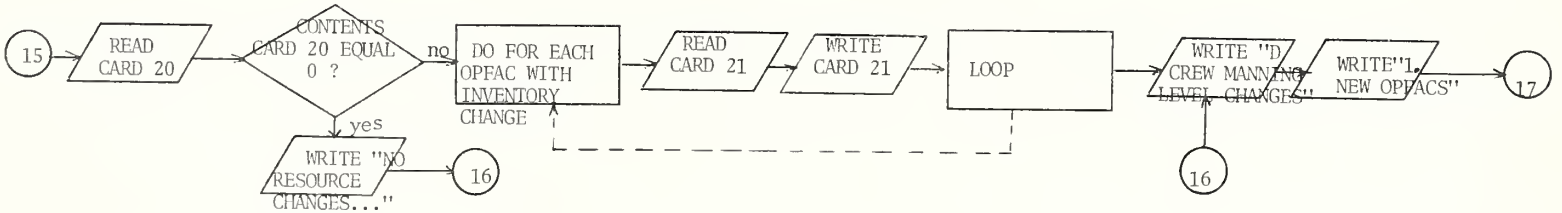


SUBROUTINE JUMPER

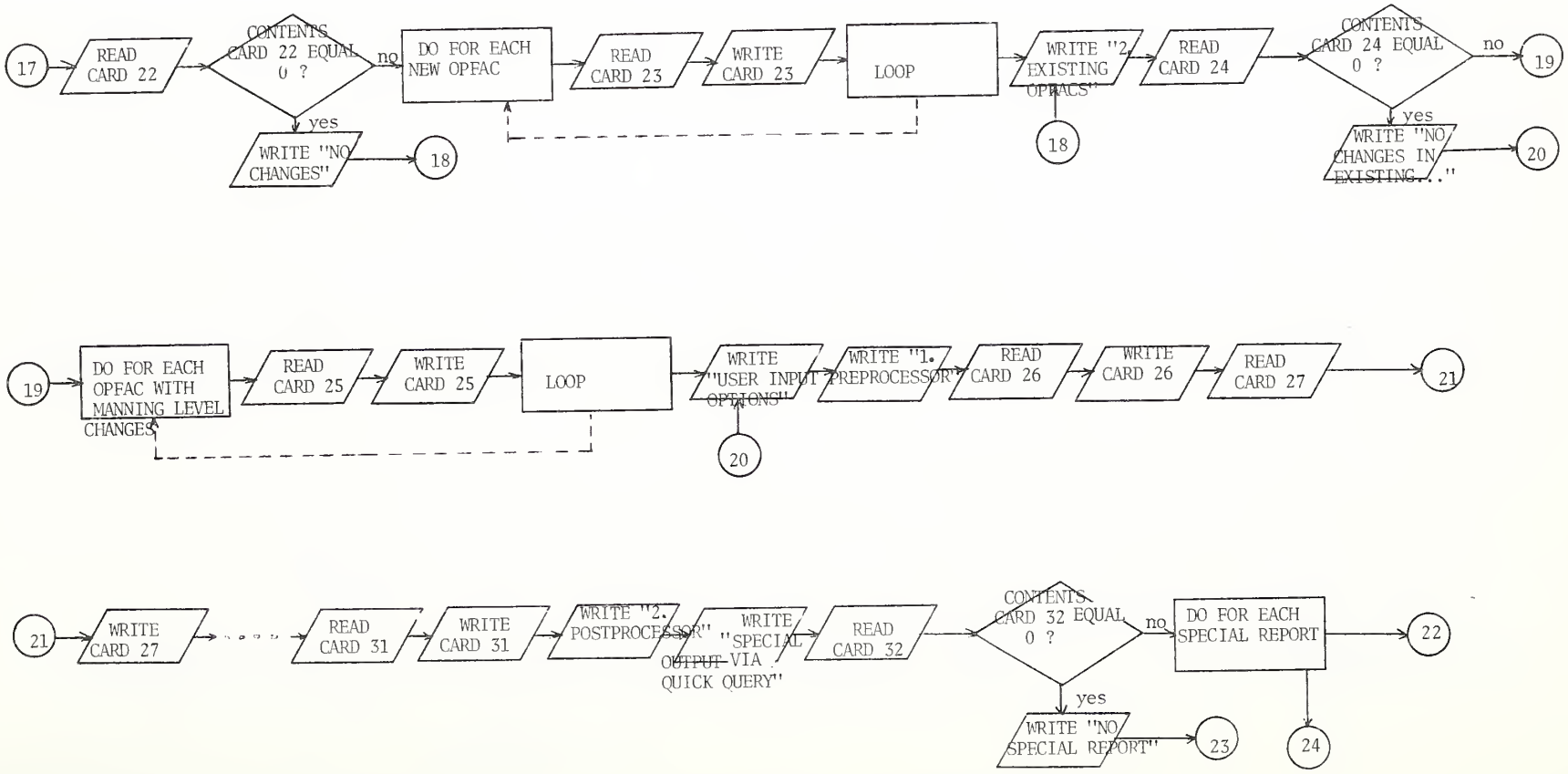



11
READ CARD 16
CONTENTS CARD 16 EQUAL 0 ?
no
WRITE "SEE INITILIZATION DATA FOR..."
DO FOR EACH RESOURCE TYPE WITH CAPABILITY CHANGE
READ CARD 17
WRITE CARD 17
LOOP
WRITE "C RESOURCE INVENTORY CHANGES"
13
WRITE "NO CAPABILITY CHANGES"
12
12

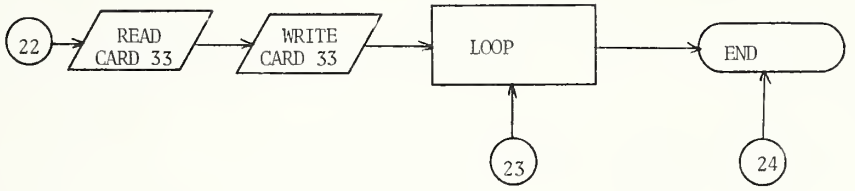

13
WRITE "1. NEW OPFACS"
READ CARD 18
CONTENTS CARD 18 EQUAL 0 ?
no
DO FOR EACH NEW OPFAC
READ CARD 19
WRITE CARD 19
LOOP
WRITE"2. EXISTING OPFACS"
15
yes
WRITE"NO NEW OPFACS"
14
14


15
READ CARD 20
CONTENTS CARD 20 EQUAL 0 ?
no
DO FOR EACH OPFAC WITH INVENTORY CHANGE
READ CARD 21
WRITE CARD 21
LOOP
WRITE "D CREW MANNING LEVEL CHANGES"
WRITE"1. NEW OPFACS"
17
yes
WRITE "NO RESOURCE CHANGES..."
16
16

SUBROUTINE JUMPER



SUBROUTINE JUMPER



DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS